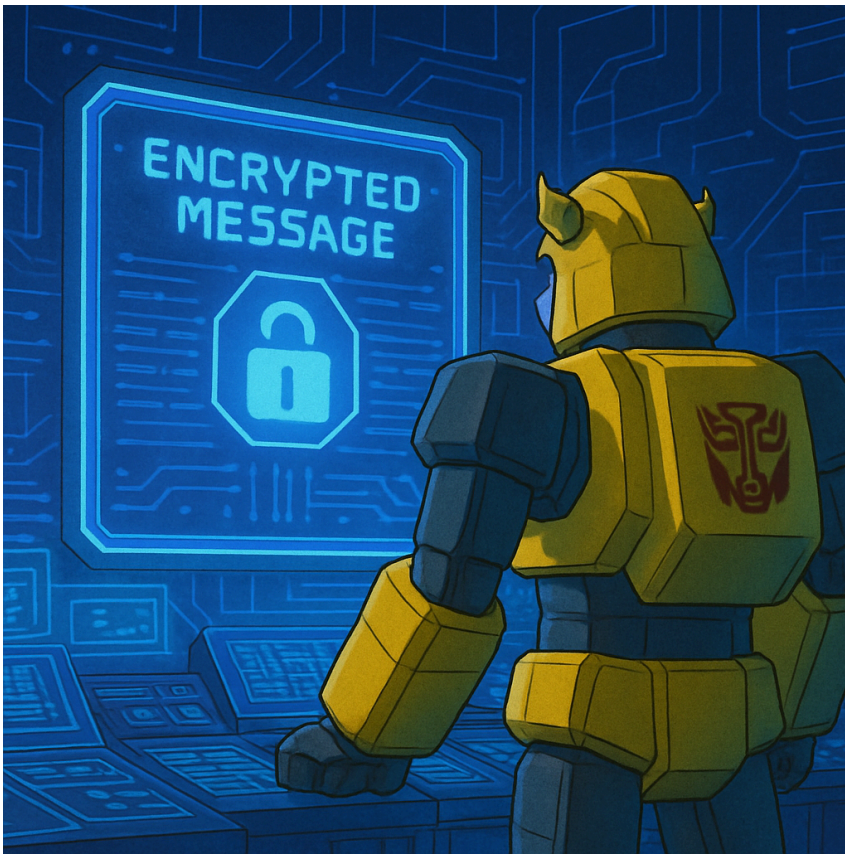
ELECTRICAL TEAM TRAINING

TASK 6

Task 6: Encrypted Message

About



Deep within the ruins of a forgotten Cybertronian outpost, a critical message has been intercepted,one that could tip the balance in the war between Autobots and Decepticons. But there's a problem:The message is encrypted in an ancient Earth language format… the humans call it **Wordle**.

Only by cracking the 5 letter code can we access the coordinates to the last known Energon reserve.

Autobot Intelligence Command is counting on you. Use your programming skills to decode the word before the Decepticons do.

Time is limited. The future of Cybertron and Earth rests on this code

Requirements

- For this task you will need to use Github. Make an account in Github and create a repository where your program will be on.

  Make sure that your repo is public and has a README file

- For each new addition and modification of the code, you must include a **Git Commit** with a message describing the change

  Eg: Add game logic, Add input handling, Fix word selection bug, Implement end game feature, etc

  Do note that we can see and will evaluate this commits

- **The task must be written using python only.** Organize the code into a well-structured folder layout. Each function should reside in its relevant and logically named file, and The main entry point of the program should be a file named **main.py**.

  Make sure that your main.py is readable and easy to navigate through by maximizing your usage of functions inside of it.

- Document your process as much as possible in your **README** file in the repo**, the README will be the first thing read within your project.** With an added bonus if you documented your analysis of this task

  Make sure that all your python files are readable, organized and commented on

- Any added features are welcomed and will be positively graded, just ensure that the required features are implemented
- **Usage of code generated by AI tools WILL be detected and you WILL be heavily penalized**

Task 6.1: Backend

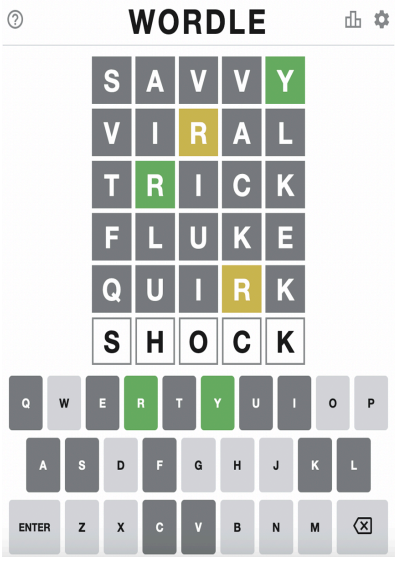
Wordle is a game about guessing a **5 letter word** in **6 attempts** or fewer. The guessed word must be a **real word** from the dictionary and not made up. You can play wordle by visiting the wordly.org site or inside discord by typing /wordle in a server

Your first task is to handle a dataset of a 5 letter word dictionary . With each round, the program must pick a **random** word within the dataset and the user must guess that word. The guess must be a **real word** from the dataset.
download a data set

Bonus: Try to minimize the time needed in navigating through the dataset

Bonus: Fetch the data from an online resource using an api and python rather than downloading the data. **Do note that this may be difficult and time consuming to do for beginners**

Task 6.2: Interface



The game interface works by taking a **5 Letter word** as input from the user and storing it inside a **5x6 Grid** like the image to the right. Each row inside the grid is a **guess** made by the user, with the first row storing the first guess. If the guess made by the user is a word

from the dataset, then the guess is **accepted** and **compared** to the random word picked by the program. If incorrect the user can make another guess in another row while the previous guess is still saved in the previous row. This continues until the user correctly guesses or used his **6 tries**

The goal of the interface is to tell the user the state he is in and the outcome of his guess, failure to do so would result in negative points

Bonus: Implement a simple gui similar to the pic above. **Do note that this may be difficult and time consuming to do for beginners as this is only a challenge to those who are already experienced with python**

Task 6.3: Game logic

For each incorrect guess, the program must indicate how close the user is to the correct guess

This is done so by letting the user know which letter is part of the correct answer, and if that letter is in its correct position or not

Bonus: Color the text to be just like the real game, green if correct letter at correct position, orange if correct letter but incorrect position and grayed out if the letter is incorrect

Task 6.4: Documentation

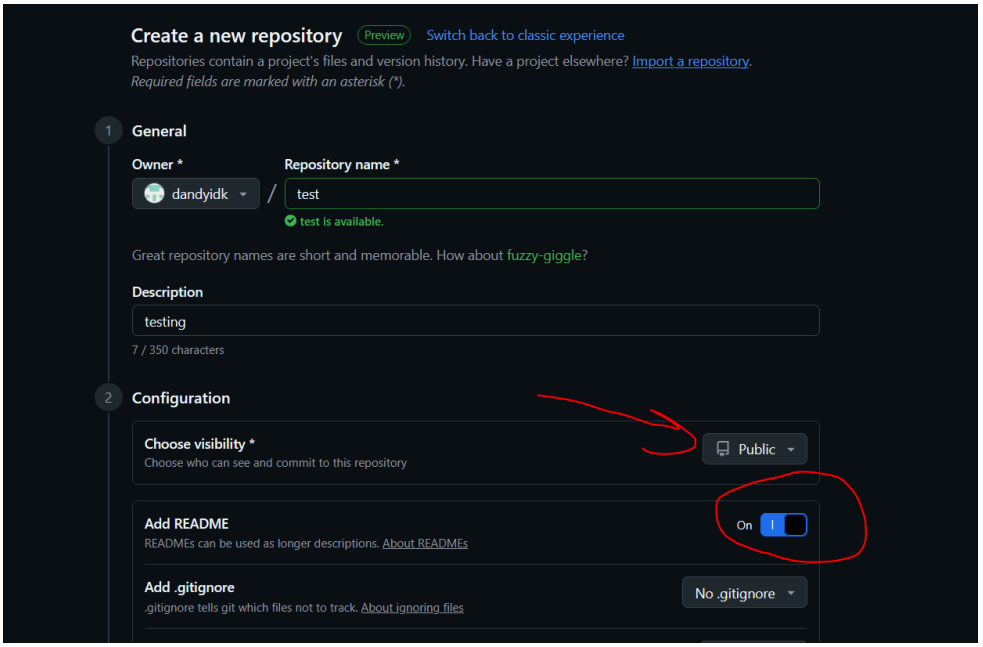
README documentations are documents that explain to the user what the project is for and how to use it.**They are the first thing to be read within the project**

For this task however you will use it to explain the **Folder structure** and **logical structure** of your code. **There is no need to explain every line of your code inside of the readme,** Just explain the logic behind it. Usage of pictures and flowcharts are encouraged

Bonus: Document your analysis of the problem along with the various solutions and strategies you considered, including those that you did not implement.

Submission

- You will submit your Github repository link:
  - Create a repository and make sure its public and includes a README file



- The Task's deadline is at Friday on 8th of august at 11:59 pm
- Submission link: https://forms.gle/rGn5wiRsfZmYCdJX9
- **Usage of code generated by AI tools WILL BE DETECTED and you WILL be heavily penalized**