## **Limitations:**

This Hangman game has some limitations, like:

**User Interface:** Hangman game is a command-line based application. This means that it lacks a graphical user interface, which can make the game less engaging for some users.

**Limited Vocabulary:** The game is limited by the vocabulary that is included in the code. This means that the game can become repetitive and predictable after multiple playthroughs.

**Single-Player Only:** Hangman is designed as a single-player game. This means that it is not suitable for multiplayer or online play, which can limit the game's appeal to some users.

**Simplistic Gameplay:** Hangman is a relatively simple game, with limited scope for complex gameplay or strategic decision-making. This can limit the game's appeal to users who prefer more complex and challenging games.

**Limited Educational Value:** While Hangman can be a fun and engaging game, it has limited educational value beyond basic vocabulary and spelling. This can limit its usefulness as a learning tool for language learners or students.

Some of these limitations are inherent to the Hangman game concept itself, rather than specific to the Python programming language.

## **Conclusion:**

In conclusion, the Hangman game is a classic word-guessing game that has been enjoyed by generations. It involves guessing a secret word by suggesting letters, with a limited number of guesses allowed. We began by choosing a list of words to be used as the secret word, and then randomly selecting one of them for each game. We then created a function that allows the user to input their guess and checks whether the letter is present in the secret word. We also created a function to display the current state of the game, including the number of remaining guesses and the letters that have been correctly guessed. In this mini-project, we have successfully implemented the Hangman game using Python programming language.