Project Report

Project Name: Hangman – Reimagined

Programming Fundamentals L1

Sir. Mohammad Nadeem

Group members:

Simal Anjum, Zainab Raza and Iqra Ahmed

**Project Brief:**

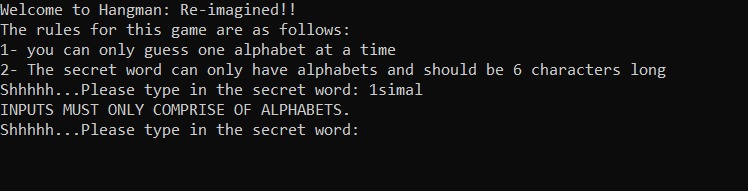
For this project we decided to write a code for Hangman (a much-loved game from our childhood) and give it a new look. The language we opted for is python.

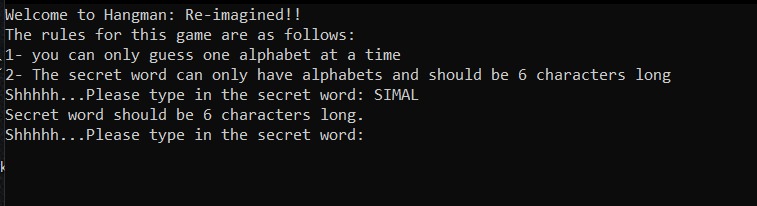
**Features:**

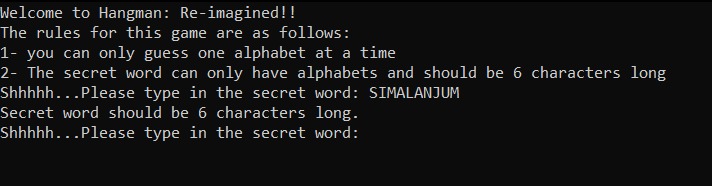
1. The secret word maybe visible to the guesser as the input for the secret word is taken just before the first guess, to avoid this we imported the “getpass” library to hide the first input i.e. “The secret word”. Example is given below.
2. The secret word can only have 6 letters, any word more or less than this limit will be discarded and the user will be asked to input again.
3. The secret word and guesses can only have alphabets if the input is a number, then the user will be asked to guess again.
4. For our program we have 3 functions: pattern, secret\_check and guess\_check.
5. In guess\_check and secret\_check we have used while loop and if else statements to implement conditions for our code.

**Test Cases:**

You can see both the inputs here for demonstration purposes.







**Here is what the code looks like when it is run:**

Welcome to Hangman: Re-imagined!!

The rules for this game are as follows:

1- you can only guess one alphabet at a time

2- The secret word can only have alphabets and should be 6 characters long

Shhhhh...Please type in the secret word:

Let's play Hangman

+---+

| |

|

|

|

|

==========

None

\_\_

What's your guess?: a

You're correct! , guess, is in the word!

+---+

| |

|

|

|

|

==========

None

\_\_\_A

What's your guess?: e

Try Again! ,guess, isn't in the word :(

+---+

| |

O |

|

|

|

==========

None

\_\_\_A

What's your guess?: i

You're correct! , guess, is in the word!

+---+

| |

O |

|

|

|

==========

None

I\_\_A

What's your guess?: o

You're correct! , guess, is in the word!

+---+

| |

O |

|

|

|

==========

None

I\_OA

What's your guess?: s

You're correct! , guess, is in the word!

+---+

| |

O |

|

|

|

==========

None

SI\_\_OA

What's your guess?: z

Try Again! ,guess, isn't in the word :(

+---+

| |

O |

| |

|

|

==========

None

SI\_\_OA

What's your guess?: x

Try Again! ,guess, isn't in the word :(

+---+

| |

O |

/| |

|

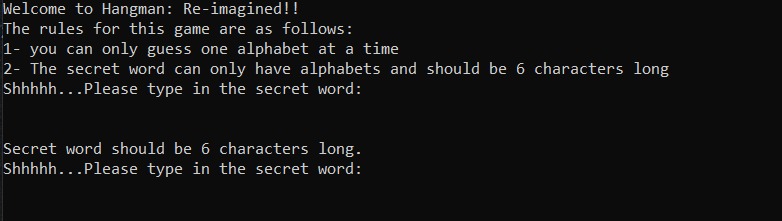
|

None

SI\_\_OA

What's your guess?:

Here is a screenshot of how the getpass library helps hide the secret word.



**Constraints:**

1. The secret word cannot have more than 6 alphabets
2. The alphabets already guessed should have been appended to a list so that the user does not make the same guess again.
3. We have the option to play again but it does not take the input for the secret message again and the input for the play again should be in capital letters.
4. Our program would have a better look if we used graphical user interface