**PYTHON MINI PROJECT**

**Topic:** Hangman Game

**Goal:** This project is a game we all have played in our past. This is the Hangman game. After starting the game user will get a word to guess in 6 tries. The tries will be displayed visually.

**Team Memebers:** Harsh Chhikara-2K18CSUN01019

Abhinav Wadhwa-2K18CSUN01142

**Code:**

**hangman code**

import import\_ipynb

import random

import words

from words import word\_list

def get\_word():

word = random.choice(word\_list)

return word.upper()

def play(word):

word\_completion = "\_" \* len(word)

guessed = False

guessed\_letters = []

guessed\_words = []

tries = 6

print("Let's play Hangman!")

print(display\_hangman(tries))

print(word\_completion)

print("\n")

while not guessed and tries > 0:

guess = input("Please guess a letter or word: ").upper()

if len(guess) == 1 and guess.isalpha():

if guess in guessed\_letters:

print("You already guessed the letter", guess)

elif guess not in word:

print(guess, "is not in the word.")

tries -= 1

guessed\_letters.append(guess)

else:

print("Good job,", guess, "is in the word!")

guessed\_letters.append(guess)

word\_as\_list = list(word\_completion)

indices = [i for i, letter in enumerate(word) if letter == guess]

for index in indices:

word\_as\_list[index] = guess

word\_completion = "".join(word\_as\_list)

if "\_" not in word\_completion:

guessed = True

elif len(guess) == len(word) and guess.isalpha():

if guess in guessed\_words:

print("You already guessed the word", guess)

elif guess != word:

print(guess, "is not the word.")

tries -= 1

guessed\_words.append(guess)

else:

guessed = True

word\_completion = word

else:

print("Not a valid guess.")

print(display\_hangman(tries))

print(word\_completion)

print("\n")

if guessed:

print("Congrats, you guessed the word! You win!")

else:

print("Sorry, you ran out of tries. The word was " + word + ". Maybe next time!")

def display\_hangman(tries):

stages = [

"""

--------

| |

| O

| \\|/

| |

| / \\

-

""",

"""

--------

| |

| O

| \\|/

| |

| /

-

""",

"""

--------

| |

| O

| \\|/

| |

|

-

""",

"""

--------

| |

| O

| \\|

| |

|

-

""",

"""

--------

| |

| O

| |

| |

|

-

""",

"""

--------

| |

| O

|

|

|

-

""",

"""

--------

| |

|

|

|

|

-

"""

]

return stages[tries]

def main():

word = get\_word()

play(word)

while input("Play Again? (Y/N) ").upper() == "Y":

word = get\_word()

play(word)

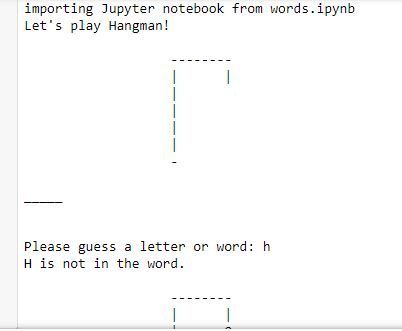
if \_\_name\_\_ == "\_\_main\_\_":

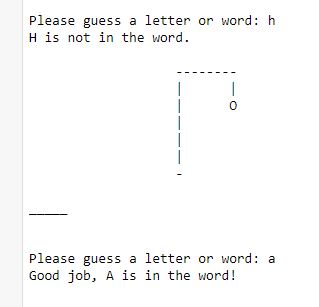
main()

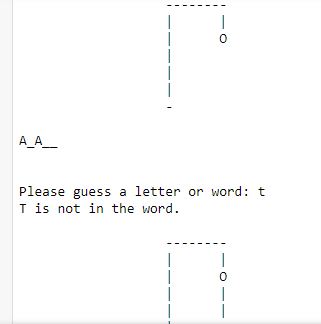
**Words Code**

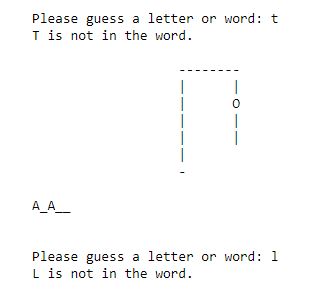
word\_list = [ 'wares', 'soup','mount','extend', 'brown','expert','tired','humidity','backpack', 'crust', 'dent', 'market','knock', 'smite', 'windy', 'coin', 'throw','silence', 'bluff', 'downfall','climb','lying','weaver', 'snob','kickoff', 'match', 'quaker', 'foreman','excite', 'thinking', 'mend', 'allergen', 'pruning','coat' , 'emerald','coherent', 'manic','multiple','square', 'upset', 'robotics', 'while','jaguar','seminary','command','cassette','draw','anchovy','scream','blush','organic','applause','parallel','trolley','pathos','origin','hang','pungent','angular','stubble', 'painted','forward','saddle','muddy', 'orchid', 'prudence','disprove','yiddish','lobbying','neuron','tumor', 'haitian','swift', 'mantel','wardrobe','consist','storied','extreme', 'payback','control','dummy', 'influx','realtor', 'detach','flake', 'consign','adjunct','stylized','weep','prepare','pioneer','tail','platoon','exercise','dummy','clap','actor', 'spark','dope', 'phrase', 'welsh', 'wall','whine','fickle', 'wrong','stamina', 'dazed','cramp','filet', 'foresee','seller','award', 'mare','uncover','drowning', 'ease', 'buttery','luxury','bigotry', 'muddy', 'photon', 'snow', 'aide','breeder', 'concoct', 'pathway', 'hearing', 'bayou', 'regimen','drain', 'bereft', 'matte', 'bill', 'medal','prickly', 'sarcasm', 'stuffy', 'allege','monopoly', 'lighter','repair', 'worship','vent', 'hybrid', 'buffet', 'lively']

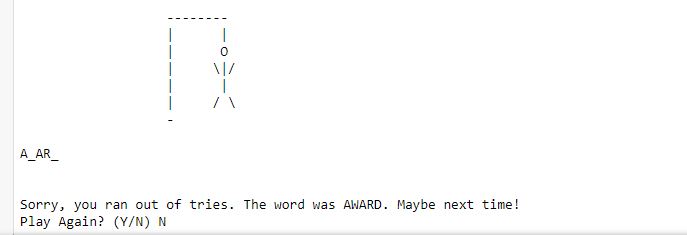
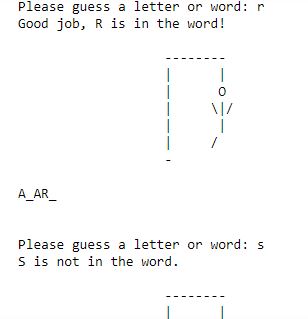
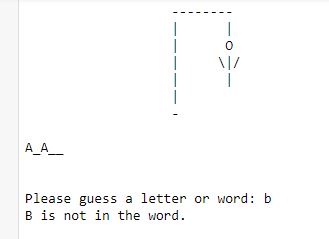
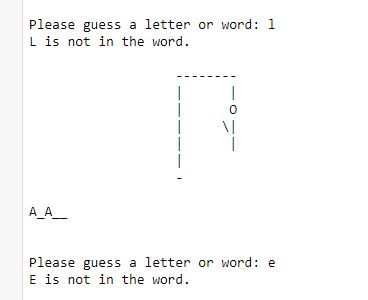
**OUTPUT:**











**My contribution to this project is the” words” code and I had some issues making the video so I’m explaining you right here.**

**I have created a python file called “Words” in which i created a function called “word\_List” which contains around 500 words which will be used in the main code. As we have imported random library, one out of these 500 words will be chosen and will be displayed on the game intro screen as \_ (underscore)or \_\_\_(dash) which will be the equal of the word .**