DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068

CERTIFICATE

This is to certify that Mr./Ms. Varun V, Vishnu S Nair, Suvik Sharma & Varsha R bearing USN: ENG19CS035, ENG19CS0363, ENG19CS0327 & ENG19CS0354 has satisfactorily completed his/her Mini Project as prescribed by the University for the III semester B.Tech. programme in Computer Science & Engineering during the year 2020-21 at the School of Engineering, Dayananda Sagar University., Bangalore.

Date: 20th November 2020

Signature of the faculty in-charge

Max Marks	Marks Obtained

Signature of Chairman
Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this mini project entitled - " Hangman Game & Desktop Notifier", has been carried out by us and it has not been submitted for the award of any degree, diploma or the mini project of any other college or university.

Varsha R-(ENG19CS0354)
Suvik Sharma-(ENG19CS0327)
V V Sai Yasasvi-(ENG19CS03246)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman, Dr. Sanjay Chitnis** ,for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to Gaurav Kumar, for providing help and suggestions in completion of this mini project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

Varun V-(ENG19CS0355)
Vishnu S Nair-(ENG19CS0363)
Suvik Sharma-(ENG19CS0327)
Varsha R-(ENG19CS0354)

TABLE OF CONTENTS

ABSTRACT

- **Python is a powerful high level machine language that uses a lot modules and functions to execute code faster & easier.**
- **Our team is demonstrating the code for a desktop notifier on a pc using python which in turn is executed by using either of the 2 modules called as :**
  - **Plyer**
  - **ToastNotifier**
- **By using these 2 modules, we can either send a desktop notification at the instant, or at a specific time given by the user.**
- **This helps us remember the tasks-to-do**

- **Another block of code that we will be executing, is that of a Hangman Game.**
- **This game uses, a lot of concepts of lists, dictionaries, importing modules, py files, conditional statements, and usage of**

INTRODUCTION

- **We will be demonstrating the working of python code and how simple fragments of code can control notification.**

Our team consists of Varsha R, Suvik Sharma, Vishnu S Nair and Varun V

Problem Statement

1. **Desktop notifier**
2. **Hangman Game**

OBJECTIVES

The main objective is to learn how the python code works, and how it uses all the basic conditional statements to achieve what we want.

Methodology

We are using the import function to import 2 different kinds of modules and functions in the modules to make sure that we solve our problem statement.

We use lists to create a new list that chooses random words to guess in the hangman game.

This is done by the random function available to us in the python module itself.

It helps us choose a function or a number or a word or a character at random.

We guess the letters and it makes a list of all the letters and make it into a list.

And then the list is printed out as the letters that we have guessed out.

Software & Hardware requirements

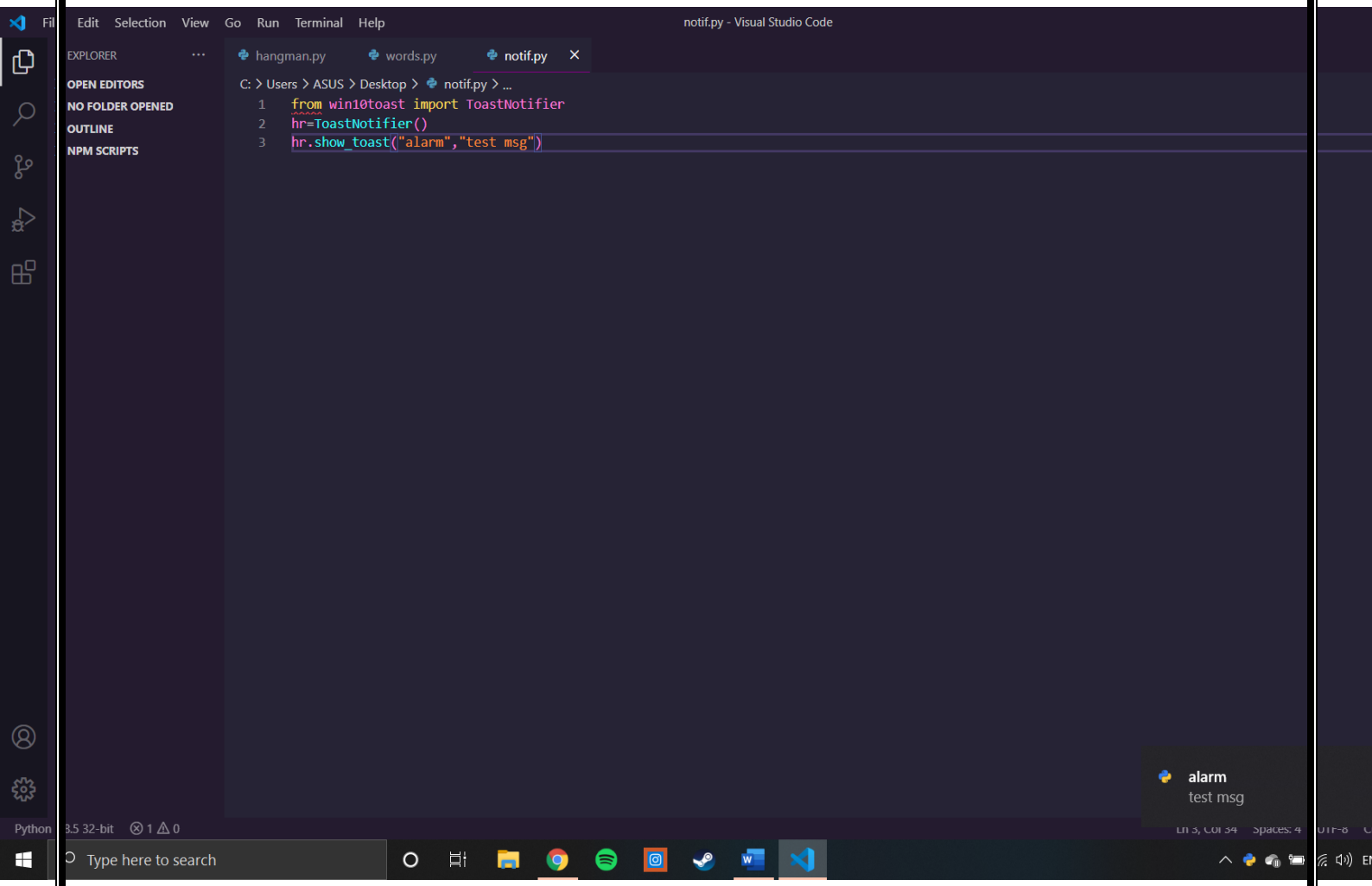
- **OS- WINDOWS 7 AND ABOVE/ UBUNTU LINUX / MAC OS**
- **PROGRAMMING LANGUAGE – PYTHON**
- **TEXT EDITOR – ATOM, VS CODE.**

Code

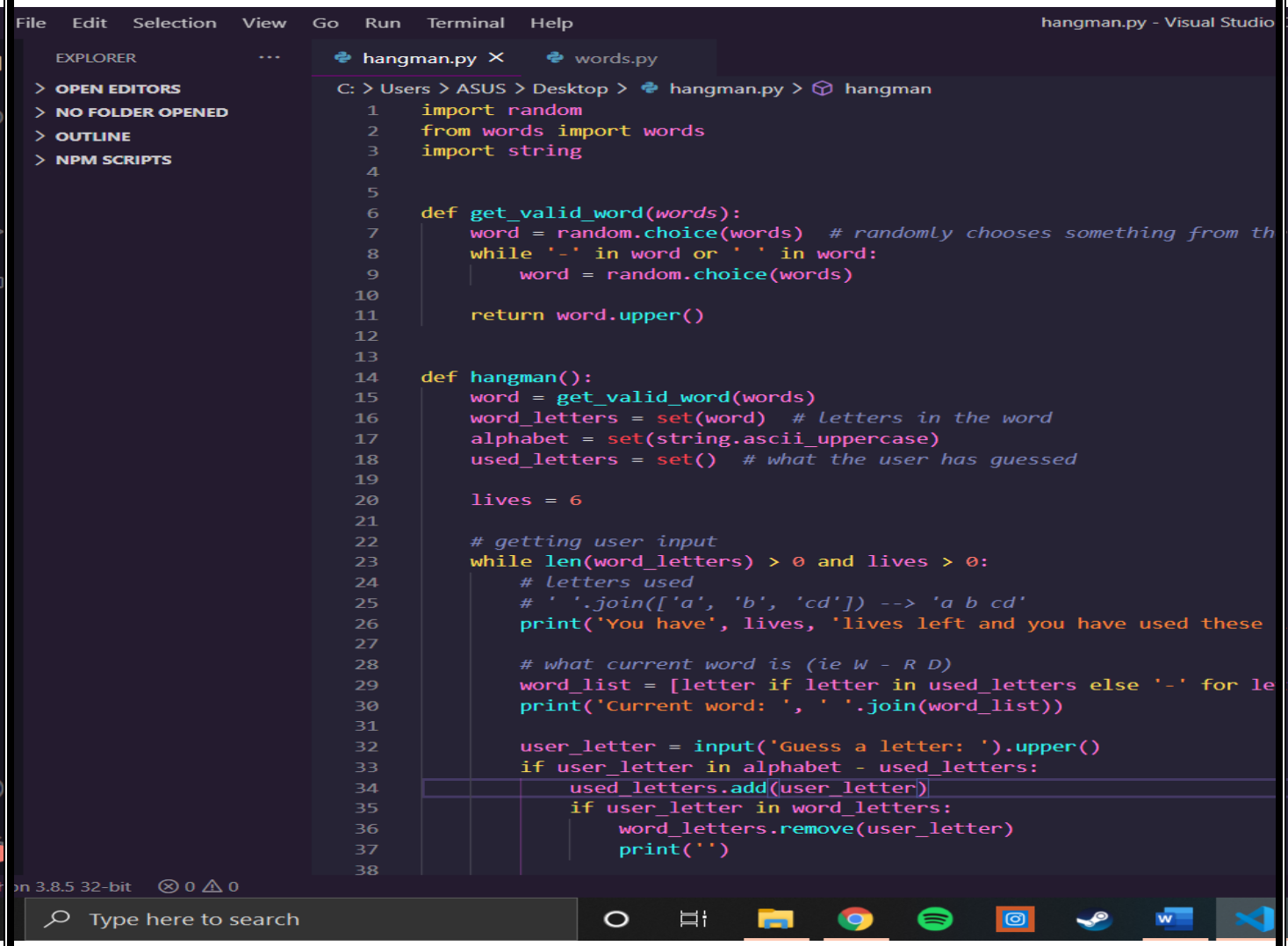
1.

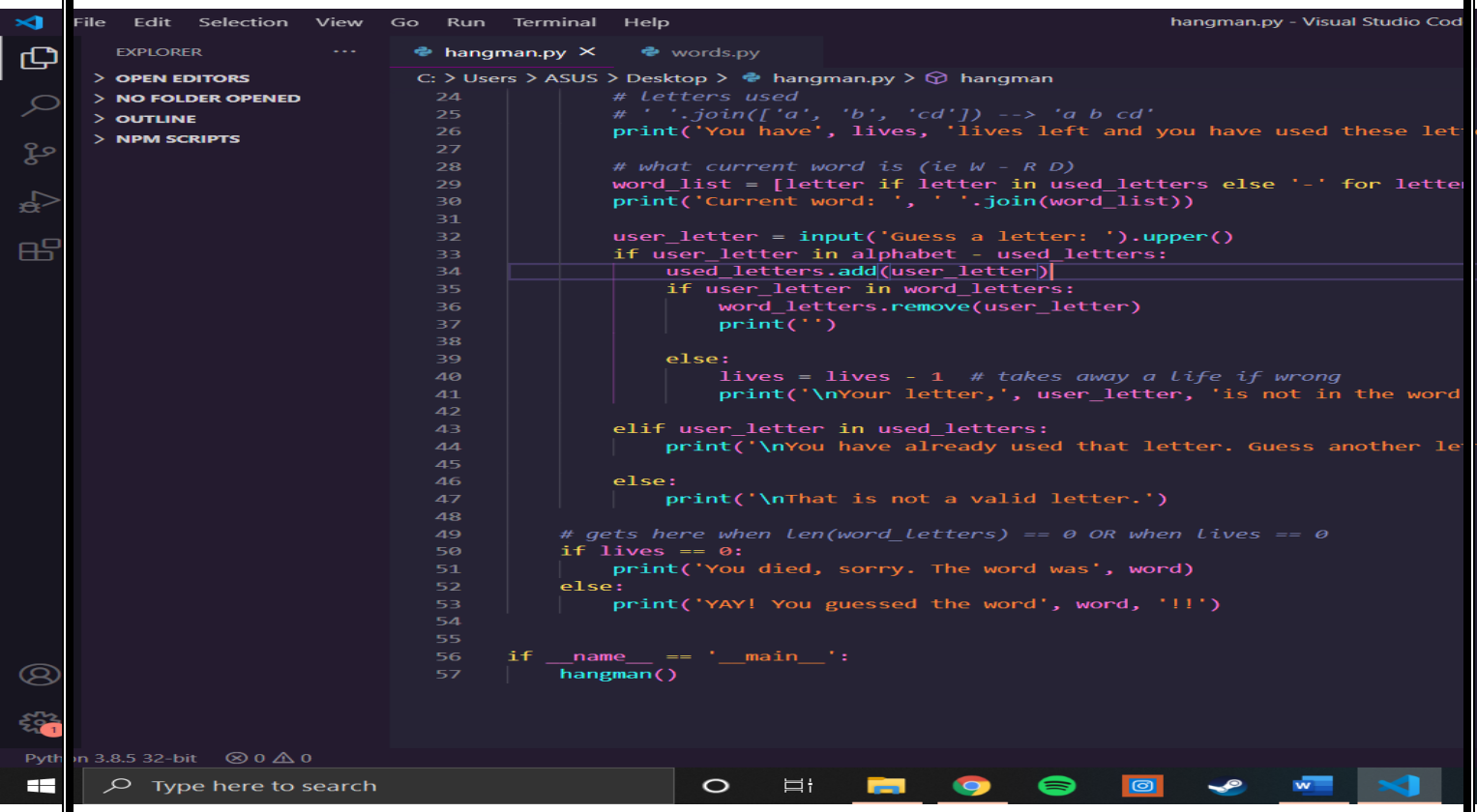
```
from win10toast import ToastNotifier
import time

while True:
    current_time =time.strftime("%H:%M:%S")
    if current_time == "9:53:00":
        print(current_time)
        break
else:
    pass

hr=ToastNotifier()
hr.show_toast("alarm","test msg")
```

RESULTS

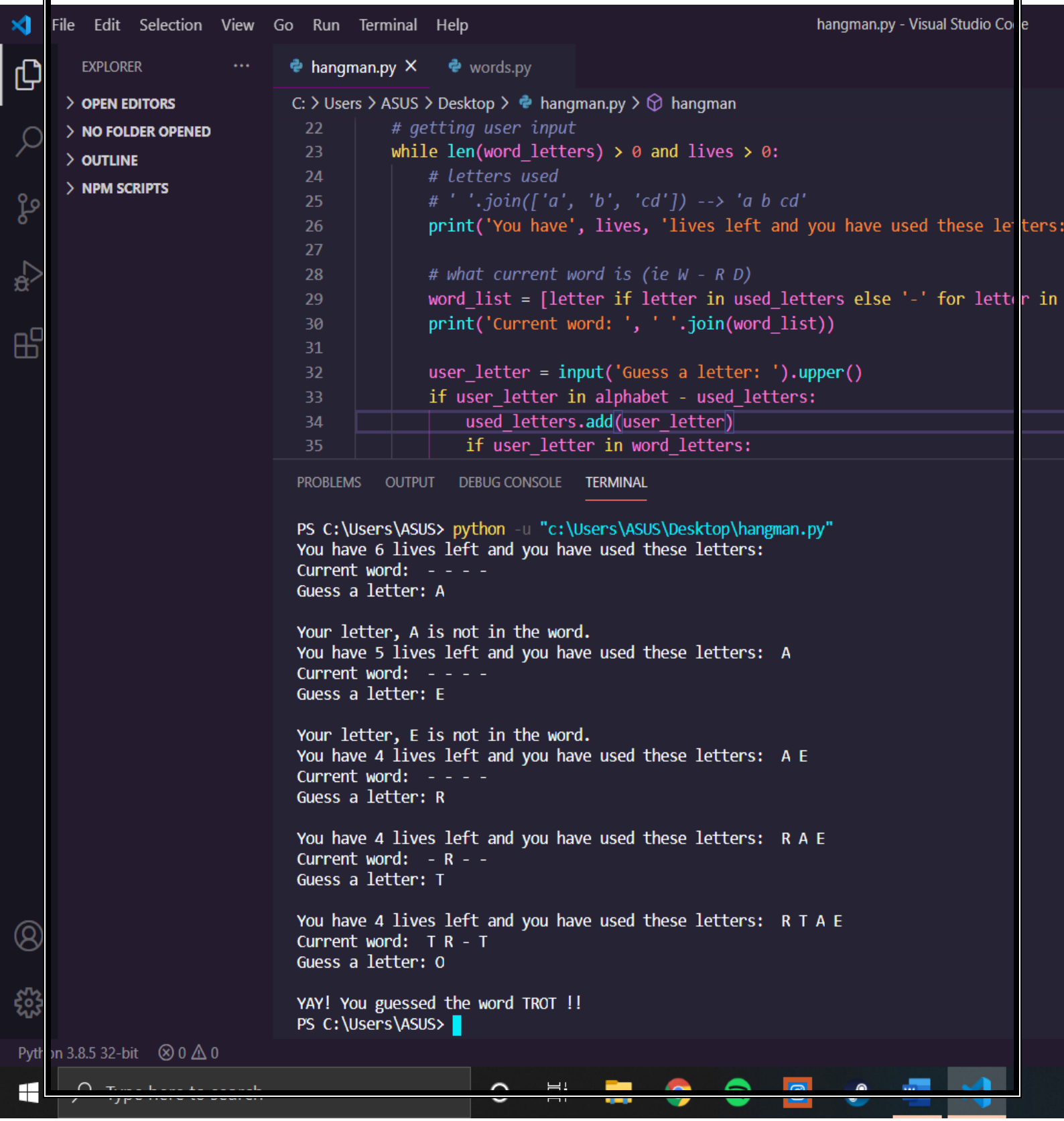


2 Hangman Game


File Edit Selection View Go Run Terminal Help
hangman.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
OUTLINE
NPM SCRIPTS
hangman.py
words.py
C: > Users > ASUS > Desktop > hangman.py > hangman
import random
from words import words
import string
def get_valid_word(words):
word = random.choice(words) # randomly chooses something from th list
while '-' in word or ' ' in word:
word = random.choice(words)
return word.upper()
def hangman():
word = get_valid_word(words)
word_letters = set(word) # letters in the word
alphabet = set(string.ascii_uppercase)
used_letters = set() # what the user has guessed
lives = 6
# getting user input
while len(word_letters) > 0 and lives > 0:
# letters used
# ' '.join(['a', 'b', 'cd']) --> 'a b cd'
print('You have', lives, 'lives left and you have used these lette
# what current word is (ie W - R D)
word_list = [letter if letter in used_letters else '-' for le ter
print('Current word: ', ' '.join(word_list))
user_letter = input('Guess a letter: ').upper()
if user_letter in alphabet - used_letters:
used_letters.add(user_letter)
if user_letter in word_letters:
word_letters.remove(user_letter)
print('')
Python 3.8.5 32-bit
0 0
Type here to search


File Edit Selection View Go Run Terminal Help
hangman.py - Visual Studio Cod
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
OUTLINE
NPM SCRIPTS
hangman.py
words.py
C: > Users > ASUS > Desktop > hangman.py > hangman
letters used
' '.join(['a', 'b', 'cd']) --> 'a b cd'
print('You have', lives, 'lives left and you have used these let ers:
# what current word is (ie W - R D)
word_list = [letter if letter in used_letters else '-' for lette in w
print('Current word: ', ' '.join(word_list))
user_letter = input('Guess a letter: ').upper()
if user_letter in alphabet - used_letters:
used_letters.add(user_letter)
if user_letter in word_letters:
word_letters.remove(user_letter)
print('')
else:
lives = lives - 1 # takes away a life if wrong
print('\nYour letter,', user_letter, 'is not in the word ')
elif user_letter in used_letters:
print('\nYou have already used that letter. Guess another le ter.'
else:
print('\nThat is not a valid letter.')
# gets here when len(word_letters) == 0 OR when lives == 0
if lives == 0:
print('You died, sorry. The word was', word)
else:
print('YAY! You guessed the word', word, '!!')
if __name__ == '__main__':
hangman()
Pyth n 3.8.5 32-bit
0 0
Type here to search

Results for Hangman :



CONCLUSION

- **We are importing 'words.py' and using hangman function to find out the word in the hangman game.**
- **Then we use conditional statements to get to the letters in the program, and make up the word.**
- **The 'words.py' has a list of words that gets chooses a word at random using the random.choice(words) that the user gets to guess at a game.**
- **In the desktop notifier we install a new module from the internet and call it from win10toast which can be installed from terminal running as an administrator.**
- **Then we import the win10toast and use the ToastNotifier() function to send notification to the computer at the required time.**

REFERENCE

- **Python Crash Course**
- **Python Course from Coursera.org**
- **YouTube videos**