INFORMATICS INSTITUTE OF TECHNOLOGY

Foundation Certificate in Higher Education

Module Code	**:**	DOC334
Module Name	**:**	Introduction to Programming II
Module leader	**:**	Mr. Sudarshana Welihinda
Semester	**:**	2
Assessment type	**:**	Individual Coursework (ICW)
Submission Date	**:**	15.12.2021

Student ID	**:**	20210076
Student Name	**:**	Fathima Nazeefa Anees
Group	**:**	B

Executive Summary

Python is an interpreted, object—oriented, high level programming language which is widely used by many programmers around the world. Moreover, this report discusses the code created for the hangman game. The codes, screenshots of the running program and the test cases are attached in the report. Furthermore, this game is a single player game, in which a player can play the hangman game where the user must guess the secret word. The user can also view the history of the past game plays and the stats of the game.

Acknowledgement

I have taken many countless efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

First and foremost, I would like to convey my special gratitude to our lecturer Mr. Nishan Saliya, who has invested his full effort in teaching us and guiding us to develop our skills in python programming, which has assisted me in my coursework.

I would also like to acknowledge with much appreciation the role of Mr. Sudarshana Welihinda, who guided us and supported us to grasp the basics on python in the first semester, which has helped me to coordinate in my coursework.

I would also like to appreciate the guidance given to us by Ms. Keerthiga Rajenthiram and Ms. Tharushi Sandamali Amarasinghe, who supported us in our improvement in python programming, by providing feedbacks, suggestions, encouragement, and advice, in our tutorial and self-study sessions. Thanks to their guidance and support.

I would also like to thank my parents for always being with me and supporting me in all situations. Last but not the least, my thanks and appreciations also go to my colleagues and to the people who have willingly helped me out with their abilities.

With sincere thanks,

Nazeefa

Table of Contents

List of Tables

List of figures

List of Acronyms

GUI	:	Graphical user interface
DB	:	Database
HTML	:	Hypertext markup language
OS	:	Operating system
SQL	:	Structured Query language

1. Introduction

This report was documented to cater to the institutional requirement of Informatics Institute of technology to evaluate the students' performance following the DOC334 module, which focuses on programming in python.

The report discusses the task, on which a python 3.x program must be created. The task and the understanding of the task provided are discussed. Furthermore, the python codes have been copied into the report, and the constructs such as packages, modules and modules have been explained. Moreover, the screenshots of the running program in various states have been included with the test cases.

Various e-resources were referred to when the program was done.

2. Task

2.1 Problem

You are to create a Python 3.x program which mimics the single-player game called "Hangman". Your program must run in the console. GUI base game will result ZERO marks. You can decide how the game menu and the console interface will look like, based on the above requirements. Since this is a console-based application, the "hangman" animation or graphics is not displayed. You must clearly state your assumptions if you have any.

Tasks to Complete

1. You must use proper Python 3.x program constructs such as packages, modules, functions, variables, data structures, etc. to develop this program.

2. The player must be able to do below tasks

• View the past game play history

  1. This is stored by using either a file base or a database
  2. If a DB is used, username must be "root" with no passwords and the server must be "localhost"
  3. You should also provide means to import/restore your current database for marking purposes. Failing to do this will result lower marks or no marks for that component.
  4. If a text file is used, the extension must be of .TXT

• A single player must be able to play this game as many times he/she likes.

  1. Players must also have the ability to exit the game as they desire at any time.

• Your program should keep track of wins and losses of each game play. This can be for the whole game play history or per session or both!

• Your program must be able to display stats like below. This can be for the whole game play history or per session.

1. Total number of games played
2. Total wins
3. Total loses

3. You can use external packages which will help you to develop the game. However,

    • Such package uses must be explained in the report.

        1. Purpose of using it and which parts were implemented with it

    • Proper instructions must be given to do the installation of such packages.

    • Links where you install/download the packages must be provided in the report

    • Failing to do above tasks will result lower marks or zero marks for those components

    • You CANNOT use ready-made game packages for this game! Such attempts will get zero marks for this whole ICW.

4. A challenge activity will be to display the result in a HTML file so you can see the past game plays in a web browser.

    • You'll get extra 10 marks for completing this task

5. You are to develop a console-based application with a suitable menu system to complete this task

2.2 Problem Understanding

It is required to develop a python 3.x program which runs in the console to mimic the hangman game. A menu system should be created which can provide the following options to the user; To play the hangman game, to view the past game history of the session or of all the games played and to view the game stats of the session or of all the games played.

- The hangman game:

  The program will generate a random word to the user from the words stored by the programmer. The word generated will be hidden with underscores when displayed to the user. The user will get a hint, and few turns to guess the word, by guessing a letter at a time. The number of turns the user receives to guess the word will be the length of the word.

  If the user guesses a correct letter, the places which contain that letter will be exposed to the user. The number of turns the user has will not exhaust

  Eg: If the word is 'APPLE' and the user guesses 'P', '_ PP _ _' will be displayed to the user.

  If the user guesses a wrong letter, the number of turns will reduce accordingly.

  The user should be able to play the game any number of times.

- The game history:

  The user should be able to view the game history of the whole game or for that session, which includes the data such as: Name, Word guessed, Turns provided, Turns used, and win/lost status
- Game stats:

  The user should be able to view the game stats for the whole game history or for that session, which includes the data of the total number of games played, total number of wins, and the total number of losses. As a challenge activity the stats must be displayed in a HTML file so that it can be viewed on a web browser.

3. Python codes

The program contains the main program, and the modules package which has all the modules and the respective functions coded into it. Few python built-in modules and external packages too were used to develop this program. The main program is Hangman.py and the modules package has the following modules; common.py, game.py, game_hist.py, game_rules.py and game_stats.py.

3.1 Main program (Hangman.py)

The main program (Hangman.py) has the word lists and the hint lists created, with the process for the menu system.

Code for Hangman.py

```
#---------------------------------IMPORT USER MODULES---------------------
-------------
import modules.common as common
import modules.game as game
import modules.game_rules as game_rules
import modules.game_hist as game_hist
import modules.game_stats as game_stats


#----------------------------------CREATE VARIABLES-----------------------
-------------
choice=0
try_again='Y'
user_name=''
Try=''
word=''
Turns=0
TurnsUsed=0
hintGiven=''
result=''

#------------------------------------CREATE LISTS-------------------------
-------------
#List of words

e_words=['cat','clock','doctor','driver','puppy','grapes','mouth',
         'soap','lipstick','garage','maths','broom','bark','nail',
         'ring','clip','root','goat','tap','park']

m_words=['butterfly','fence','hangar','ketchup','singer','kiwi',
         'moustache','kettle','grater','stethoscope','skateboard',
         'biology','sixty','ireland','backpack','fireworks','rocket',
         'yellow','peach','eleven']

h_words=['rhinoceros','technician','cheeseburger','croissant',
         'masseuse','intestine','saxophone','tambourine','wheelchair',
         'escalator','france','lymph','cryptography','hajj','quiz',
         'fizzy','absurd','jazz','cozy','jogging']

#List of hints

e_hint=['an animal','used to see the time','an occupation',
        'an occupation','an animal','a fruit','part of the face',
        'used to clean the body','a makeup item',
        'vehicles are parked here','subject','used to clean the house',
        'sound of an animal','used to hang something on the wall',
        'an accessory','an accessory','part of a tree','an animal',
        'get water from','place where kids play']


m_hint=['an insect','a guard for an outdoor area','place where aeroplanes
are parked',
        'used as a relish','an occupation','a fruit','visible on the
face',
        'found in the kitchen','found in the kitchen','used by doctors',
        'used to play','a subject','a number','a country','taken to
school',
        'decorates the sky','a type of firework','a color','a fruit','a
number']


h_hint=['an animal','an occupation','food','food','an occupation',
         'part of the human body','musical instrument','musical
instrument',
         'used in the hospital','a staircase','a country',
         'a fluid containing white blood cells','a field of study','a
festival/ritual',
         'a question-answer session','type of drink','illogical','a music
genre',
         'comfort','exercise']
```

```
#------------------------------------PROCESS------------------------------
-------------

#clear console
common.clearConsole()

#drops the perSession History table
game_hist.sessionTable()

#Welcomes the user
print ('\nWELCOME TO THE HANGMAN GAME!!\n')
common.symbol()
print ('\n')

user_name = input ('Enter your name please: ').upper()

common.clearConsole()

print ("\nHi",user_name,"! \nHope you are doing good!!\n\nLET'S START\n")
print ("What would you like to do?\n")



#MENU SYSTEM

while try_again == 'Y' :

    #display the menu and get the choice
    choice=common.menu_choice()

    while True :
        Try = 'Y'

        #process for 1st choice
        if choice == 1 :

            #works when try is 'Y',i.e. if the user wants to replay the
game
            while Try == 'Y' :

                common.symbol()

                #plays the game
                #gets the necessary varaiables from the respective
functions
```

```
                date,Time = common.time_date()
                mode = game.game_mode()
                word = game.random_word(e_words,m_words,h_words)
                game.hide_word()
                Turns,TurnsUsed,hintGiven,result =
game.find_word(e_hint,m_hint,h_hint)

                #Adds the data to the respective tables in the databases
                game_hist.datainput(date,Time,user_name,mode,word,Turns,Tu
rnsUsed,hintGiven,result)
                game_hist.sessionData(date,Time,user_name,mode,word,Turns,
TurnsUsed,hintGiven,result)

                #Asks the user if he/she needs to replay the game before
going to the main menu
                while True:
                    Try = input('To play again, enter "Y": ').upper()
                    common.Exit(Try)
                    common.clearConsole()

                    #welcomes the user back if the user needs to play
again

                    if Try == 'Y' :
                        print('\nHi again!!\n')
                        break

                    #if the input is invalid or the user doesnt want to
play again

                    else:
                        break


        #process for 2nd choice
        elif choice == 2 :
            common.symbol()

            #displays game rules
            game_rules.rules()


        #process for 3rd choice
        elif choice == 3 :

            #works when try is 'Y',i.e. if the user wants to view other
history
```

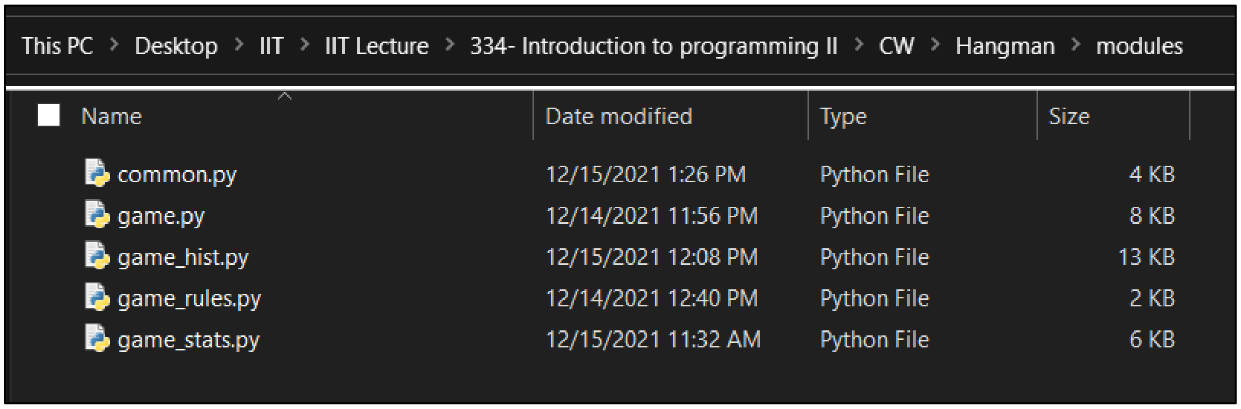
```
            while Try == 'Y' :

                common.symbol()

                #displays past game history
                game_hist.view_hist()

                #Asks the user if he/she needs to view other history
                while True:
                    Try = input('To view other history, enter "Y":
').upper()

                    common.Exit(Try)
                    common.clearConsole()

                    #welcomes the user back
                    if Try == 'Y' :
                        print('\nHi again!!\n')
                        break

                    #if the input is invalid or the user doesnt want to
view again

                    else:
                        break


        #process for 4th choice
        elif choice == 4 :
            #works when try is 'Y',i.e. if the user wants to view other
stats
            while Try == 'Y' :

                common.symbol()

                #displays stats
                game_stats.html()

                #Asks the user if he/she needs to view other stats
                while True:
                    Try = input('To view other stats, enter "Y":
').upper()

                    common.Exit(Try)
                    common.clearConsole()

                    #welcomes the user back
                    if Try == 'Y' :
                        print('\nHi again!!\n')
                        break

                    #if the input is invalid or the user doesnt want to
view again
                    else:
                        break



        #exits the program if choice is 99
        elif choice == 99 :
            exit()

        #process if the choice was invalid
        else :
            print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
            break
        print()


        #process to try again
        while True :
            try_again = input('To exit enter "N",To return to the main
menu enter "Y": ').upper()
            common.Exit(try_again)
            common.clearConsole()

            #exits if the user enters 'N'
            if try_again == 'N':

                #clears the game history per session table
                game_hist.sessionTable()
                exit()

            #displays menu if the user enters 'Y'
            elif try_again == 'Y' :
                choice=common.menu_choice()
                break

            #if the user input is invalid
            else :
                print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
```

3.2 Package (modules)

The modules package contains 5 modules which will be imported to the main program. The functions were put into modules according to the task which will be completed by the respective functions with reference to the menu.

The modules are as follows:

- common.py
- game.py
- game_hist.py
- game_rules.py
- game_stats.py.



This PC › Desktop › IIT › IIT Lecture › 334- Introduction to programming II › CW › Hangman › modules

Name	Date modified	Type	Size
common.py	12/15/2021 1:26 PM	Python File	4 KB
game.py	12/14/2021 11:56 PM	Python File	8 KB
game_hist.py	12/15/2021 12:08 PM	Python File	13 KB
game_rules.py	12/14/2021 12:40 PM	Python File	2 KB
game_stats.py	12/15/2021 11:32 AM	Python File	6 KB

Figure 1(3.2.1): user-defined modules

3.2.1 common.py

This module contains 7 functions that will be commonly used for the main program as well as the other modules.

The functions are as follows:

- menu()
  This function will print the menu to the user

- menu_choice()
  This function will get the choice from the user and will return the choice

- is_letter(value)
  This function will check if the user input is an alphabet
  A string value is taken as a parameter and a Boolean value will be returned

- symbol()
  This function will print the hangman symbol. This function is used to improve the user friendliness of the program

- Exit()
  This function will exit from the program if the value taken as parameter is “EXIT”

- time_date()
  This function will get and return the current date and time

- clearConsole()
  This function will clear the console. This function is used to improve the user experience when the user is running the program

Code for common.py

```
#---------------------------------IMPORT USER MODULES---------------------
-------------
import modules.common as common
import modules.game_hist as game_hist


#----------------------------------DEFINING FUNCTIONS---------------------
-------------
#---------------------------------------FUNCTION 1-------------------------
-------------

def menu() :
    '''This function will print the menu to the user'''

    print()
    common.symbol()
    print('1.  Play Hangman game',
          '\n2.  Rules to play hangman game',
          '\n3.  View the past game history',
          '\n4.  View Stats',
          '\n99. Exit',
          '\n\n\u25CFIF YOU WANT TO EXIT THE GAME AT ANY TIME, TYPE
"EXIT"\n\n')


#---------------------------------------FUNCTION 2-------------------------
-------------

def menu_choice():
    '''
    This function will get the choice from the user

    Returns:
        choice: The number which is the choice chosen by the user from the
menu

    '''

    while True:
        c = ''
        choice = 0
        try :
            #gets the choice from the user
            common.menu()
            c=input('Enter your choice: ')
            common.clearConsole()
            common.Exit(c)
            choice=int(c)
            break

        except ValueError :
            #if input is invalid raises an error
            print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
            continue

    return choice


#-------------------------------------FUNCTION 3-------------------------
-------------

def is_letter(value):
    '''
    This function will check if the user input is an alphabet

    Parameters:
        value: a string value which is checked if it is an alphabet

    Returns:
        bool(value in alpha): Contains a Boolean value True or False
    '''

    alpha=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p'
,
           'q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F'
,
           'G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'
,
           'W','X','Y','Z']
    return(bool(value in alpha))


#-------------------------------------FUNCTION 4-------------------------
-------------
```

```
def symbol():
    '''This function will print the hangman symbol'''

    print("""
                    --------
                    |      |
                    |      O
                    |     \\|/
                    |      |
                    |     / \\
                   ---
                    """)


#--------------------------------------FUNCTION 5-------------------------
-------------

def Exit(val):
    '''
    This function will exit from the program

    Parameters:
        val: the string value to be checked if it is 'EXIT'

    '''
    #program will exit if val is 'EXIT'
    if val.upper()=='EXIT':
        game_hist.sessionTable()
        exit()


#--------------------------------------FUNCTION 6-------------------------
-------------

def time_date() :
    '''
    This function will get the current date and time

    Returns:
        date: This contains the current date
        Time: This contains the current time


    '''
    #importinng built-in modules
    import datetime
    import time

    #getting the current date and time according to the needed format
    date = datetime.date.today().strftime('%d %b %Y')
    Time = time.strftime('%H:%M')

    return date,Time


#--------------------------------------FUNCTION 7-------------------------
-------------

def clearConsole() :
    '''This function will clear the console'''

    #importing built-in modules
    import os

    #process to clear the console
    command = 'clear'
    if os.name in ('nt', 'dos'):
        command = 'cls'
    os.system(command)
```

3.2.2 game.py

This module contains 6 functions that will function if the user wishes to play the hangman game.

The functions are as follows:

- game_mode()
  This function will get the game mode the user wishes to play and will return the mode

- random_word(word_1,word_2,word_3)
  This function will get a random word from the lists according to the game mode and will return the word

- hide_word()
  This function will hide the word with underscores

- find_index(word,guess)
  This function will find the index of guess in word and will return the list which contains the index/indexes

- get_hint(hint_1,hint_2,hint_3)
  This function will display a hint for the respective word

- find_word(hint_1,hint_2,hint_3)
  This function will allow the user to guess the word and will return the Turns, Turns used, Hint and the result

Code for game.py

```
#---------------------------------IMPORT MODULES--------------------------
--------
#user modules
import modules.common as common

#built-in modules
import random

#Assigning variables
mode=''
word=''
hidden_word=''
hint=''
rand=0
hintGiven=''

#Creating lists
Index=[]
hidden_word_list=[]
guessed=[]



#----------------------------------DEFINING FUNCTIONS---------------------
-------------
#--------------------------------------FUNCTION 1-------------------------
-------------
def game_mode():
    '''
    This function will get the game mode the user wishes to play

    returns:
        Mode : Mode is a string value which contains the game mode chosen
by the user

    '''

    #Defining global variables
    global mode

    #Assigning it an empty string
    mode=''
    #Getting the mode from the user
    while mode == '' :
        print ('\nWhich mode would you like to play?')
        mode = input('Enter "E" for easy, "M" for medium and "H" for hard 
mode: ').upper()

        common.clearConsole()
        common.Exit(mode)

        if mode == 'E' or mode == 'M' or mode == 'H':
            if mode== 'E' :
                Mode = 'EASY'
            elif mode == 'M' :
                Mode = 'MEDIUM'
            elif mode == 'H' :
                Mode = 'HARD'
        else:
            mode = ''
            print ('\n\nInvalid input! Please try again\n')
    return Mode


#--------------------------------------FUNCTION 2-------------------------
-------------

def random_word(word_1,word_2,word_3):
    '''
    This function will get a random word from the lists according to the 
game mode

    Parameters:
        word_1: A list of words
        word_2: A list of words
        word_3: A list of words

    Returns:
        word: The string value which contains the random word
    '''

    #Defining global variables
    global mode,word,hidden_word,rand

    #getting a random number from 0-20
    rand=random.randrange(0,20)
    #getting the random word from the respective list
    if mode == 'E':
        word = word_1[rand].upper()
    if mode == 'M':
        word = word_2[rand].upper()
    if mode == 'H':
        word = word_3[rand].upper()


    return word


#--------------------------------------FUNCTION 3-------------------------
-------------

def hide_word():
    '''This function will hide the word with underscores'''

    #Defining global variables
    global hidden_word_list,word

    #Creating an empty list
    hidden_word_list=[]

    #Replaces the letter with underscores
    for i in range(len(word)):

        #checks if the character is space
        if ord(word[i]) == 32:
            hidden_word_list.append(' ')

        else:
            hidden_word_list.append('_')


#--------------------------------------FUNCTION 4-------------------------
-------------

def find_index(word,guess):
    '''
    This function will find the index of guess in word

    Parameters:
        word: The string value in which the index of guess should be found
        guess: The string value to get the index of
    Returns:
        Index: A list which contains the index/indexes of guess in word
    '''

    #Defining global variables
    global Index

    #Creating an empty list
    Index = []

    #Getting the index of guess
    for i in range(len(word)):

        if guess == word[i] :

            #Appending the index to the list
            Index.append(i)

    return Index


#--------------------------------------FUNCTION 5-------------------------
-------------

def get_hint(hint_1,hint_2,hint_3):
    '''
    This function will display  a hint for the respective word

    Parameters:
        hint_1: A list of hints
        hint_2: A list of hints
        hint_3: A list of hints
    '''

    #defining global variables
    global mode,hint_count,hintGiven,hint

    #checks if user has already taken a hint
    if hint_count==1:

        #asks the user if he/she needs a hint
        need_hint = input('If you need a hint, type "YES": ')

        common.Exit(need_hint)
        if need_hint.upper() == 'YES':
            hintGiven='YES'

            #makes hint_count=0, indicating the user has already taken a
hint
            hint_count = 0

            #gets the hint according to the respective word
            if mode == 'E':
                hint = hint_1[rand].upper()
            if mode == 'M':
                hint = hint_2[rand].upper()
            if mode == 'H':
                hint = hint_3[rand].upper()

        else:
            hintGiven='NO'
            hint=''

        #if a hint is provided, prints the hint
        if hint:
            print('hint: ',hint)

    #if hint is already given, prints the hint
    elif hint_count==0:
        hintGiven='YES'
        print('hint: ',hint)


#--------------------------------------FUNCTION 6-------------------------
-------------

def find_word(hint_1,hint_2,hint_3):
    '''
    This function will allow the user to guess the word

    Parameters:
        hint_1: A list of hints
        hint_2: A list of hints
        hint_3: A list of hints

    Returns:
        Turns: Contains the number of turns provided
        TurnsUsed: Contains the number of turns used
        hintGiven: Contains a yes/no stating if a hint was provided or not
        result: Contains win/loss stating if the user won or lost in the
game

    '''

    #Assigning global variables
    global Index,guessed,word,hidden_word_list,hint_count

    #Assigning variables
    tries = len(word)
    Turns=len(word)
    stop = 'no'
    guessed=[]
    hint_count=1
    hidden_word = ''.join(hidden_word_list)
    result=''


    print()

    #Prints the number of letters the word has
    print('The word has',len(word),'letters')

    while stop == 'no':

        #Checks if the word guessed by the user matches the exact word
        if hidden_word == word:

            #Turns used is the number of turns provided minus the tries
remaining
            TurnsUsed=Turns-tries
            tries = 0
            print()

            #displays the congratulations message
            print('Congratulations!!\nYou have guessed the word\n\nThe
word is: ',word)
            result = 'WIN'
            stop = 'yes'

        #checks if all the tries have exhausted
        elif tries == 0:
            print()
            #displays the message that the user has lost
            print('Sorry you have run out of tries!\n\nThe word is:
',word)
            result = 'LOSS'
            stop = 'yes'
            TurnsUsed=Turns-tries

        #if the user hasnt won or lost the game continues
        else:

            print()

            #the hidden word with underscores is displayed with a space
            print("  ",' '.join(hidden_word))
            print()

            get_hint(hint_1,hint_2,hint_3)

            #a letter is taken from the user as a guess
            guess=input('Enter the letter: ').upper()
            print()

            common.Exit(guess)

            #checks if the letter is already guessed
            if guess in guessed:
                print('You have already guessed this letter, letter:
',guess)

                #a try is exhausted
                tries -= 1
                print('You have',tries,'tries remaining')

            #checks if the input is not a letter, or if it is a blank
value
            elif common.is_letter(guess) == False or guess == '' :
                print('Invalid input')

            #checks if the guessed letter is not in the word
            elif guess not in word:
                print('Wrong guess!')

                #the guessed letter is appended to the list of guessed
letters
```

```
                guessed.append(guess)

                #a try is exhausted
                tries -= 1

                print('You have',tries,'tries remaining')

            #checks if the guessed letter is in the word
            elif guess in word:
                print('Correct guess!')

                #the guessed letter is appended to the list of guessed
letters
                guessed.append(guess)

                #finds the index of the guessed letter in the word
                I=(find_index(word,guess))

                #replaces the underscore with the respective guessed
letter
                for i in (I):
                    hidden_word_list[i]=guess
                hidden_word = ''.join(hidden_word_list)

                if hidden_word == word:
                    print()
                else:
                    print('You have',tries,'tries remaining')

    return Turns,TurnsUsed,hintGiven,result
```

3.2.3 game_hist.py

This module contains 8 functions that will function if the user wishes to view the game history. The functions are as follows:

- open_db()
  This function will open the database and check the connectivity of the database and will return the database connection

- datainput(val1,val2,val3,val4,val5,val6,val7,val8,val9)
  This function will add the details which are taken as parameters into the database

- sessionTable()
  This function will delete the existing table which records the game history per session

- sessionData(val1,val2,val3,val4,val5,val6,val7,val8,val9)
  This function will create a table and add the details which are taken as parameters in the table

- choice1()
  This function will display the game history for the session

- choice2()
  This function will display the whole game history

- choice3()
  This function will display the game history of the selected player

- view_hist()
  This function will display a mini menu for the user to select from which data the user needs to view

Code for game_hist.py

```
#---------------------------------IMPORT MODULES--------------------------
--------
#user modules
import modules.game_hist as game_hist
import modules.common as common

#built-in modules
import mysql.connector
import texttable


#---------------------------------DEFINING FUNCTIONS---------------------
-------------
#---------------------------------------FUNCTION 1------------------------
-------------

def open_db():
    '''
    This function will open the database and check the connectivity of the
database

    Returns:
        db: contains the database connection
    '''

    from mysql.connector import Error


    try_again='Y'
    while try_again=='Y':

        #Exception handling
        try:
            #opening the database connection with a dictionary
            conDict={'host':'localhost',
                    'database':'hangman',
                    'user':'root',
                    'password':''}
            db = mysql.connector.connect(**conDict)

            #loop is broken if the database is connected
            if db.is_connected():
                break
        #If an error is encountered,it displays the error
        except Error as e:
            print('You have encountered an error')

            #prints the error
            print(e)

            print('\nPlease connect to the database properly and try 
again')
            print('You can follow the instructions given in the report on 
chapter 3.5 page 46')


            print()

            #Asks the user if he/she wants to try again after connecting 
to the db properly
            while True:
                try_again=input('Enter "Y" to try again or you can exit 
the program by typing "EXIT": ').upper()
                common.clearConsole()

                #if he wants to try again
                if try_again=='Y':
                    break

                #exits if he wants to exit
                if try_again=='EXIT':
                    common.clearConsole()
                    raise exit()

                #if input is invalid, asks the user to input a valid value
                else:
                    print()
                    print('\nINVALID INPUT!!\nPlease enter a valid 
value!!\n')

        if try_again=='EXIT':
            common.Exit(try_again)
            break

    return db
```

```
#--------------------------------------FUNCTION 2-------------------------
-------------

def datainput(val1,val2,val3,val4,val5,val6,val7,val8,val9):
    '''
    This function will add the details into the database

    Parameters:
        val1: a string value which contains the date
        val2: a string value which contains the time
        val3: a string value which contains the name
        val4: a string value which contains the mode
        val5: a string value which contains the word
        val6: an int value which contains the number of turns
        val7: an int value which contains the number of turns used
        val8: a string value which contains a 'YES' or 'NO' stating if a
hint was provided
        val9: a string value stating if it is a 'WIN' or 'LOSS'

    '''
    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor = db.cursor()

    #executing SQL query to insert data into the tables
    query='INSERT INTO playerinfo
(date,time,name,mode,word,turns,turnsUsed,hintGiven,winORloss)VALUES(%s,%s
,%s,%s,%s,%s,%s,%s,%s)'
    val=(val1,val2,val3,val4,val5,val6,val7,val8,val9)

    cursor.execute(query,val)

    #commits the change
    db.commit()

    #disconnects from the server
    db.close()


#--------------------------------------FUNCTION 3-------------------------
-------------

def sessionTable():
    '''This function will delete the existing table which records the game
history per session'''
    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor = db.cursor()

    #executing SQL query to delete all the data from the table
    cursor.execute('DELETE FROM playerInfo_perSession')

    #commits the change
    db.commit()

    #disconnects from the server
    db.close()


#---------------------------------------FUNCTION 4-------------------------
-------------

def sessionData(val1,val2,val3,val4,val5,val6,val7,val8,val9):
    '''
    This function will create a table and add the details in the table

    Parameters:
        val1: a string value which contains the date
        val2: a string value which contains the time
        val3: a string value which contains the name
        val4: a string value which contains the mode
        val5: a string value which contains the word
        val6: an int value which contains the number of turns
        val7: an int value which contains the number of turns used
        val8: a string value which contains a 'YES' or 'NO' stating if a
hint was provided
        val9: a string value stating if it is a 'WIN' or 'LOSS'
    '''
    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor = db.cursor()

    #executing SQL query to alter the table by making the auto increment
to 1
```

```
    cursor.execute('ALTER TABLE playerInfo_perSession AUTO_INCREMENT = 1')

    #commits the change
    db.commit()

    #Executing the SQL query to insert data into the table
    query='INSERT INTO playerInfo_perSession
(date,time,name,mode,word,turns,turnsUsed,hintGiven,winORloss)VALUES(%s,%s
,%s,%s,%s,%s,%s,%s,%s)'
    val=(val1,val2,val3,val4,val5,val6,val7,val8,val9)

    cursor.execute(query,val)

    #commits the change
    db.commit()

    #disconnects from the server
    db.close()


#--------------------------------------FUNCTION 5-------------------------
-------------

def choice1():
    '''This function will display the game history for the session'''

    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor = db.cursor()

    #executing SQL query to display all the data from the table
    cursor.execute('SELECT * FROM playerinfo_persession')

    #fetching the data using fetchall() method
    data=cursor.fetchall()

    #counting the number of rows
    row=cursor.rowcount

    #prints 'Sorry there are no entries' if there are no rows found
    if row == 0:
        print('Sorry, There are no entries')
    #displays the data if there are rows found
    else:

        #organizing the data in table format using the texttable module
        table=texttable.Texttable()

        #Creating the heading names for the table
        headings=['ID','Date','Time','Name','Mode','Word','Turns','TurnsUs
ed','Hint','Status']
        table.header(headings)

        #inserts the rows of data into the table
        for row in data:
            table.add_row(row)

        #adjusting column width
        table.set_cols_width([5,15,7,15,8,15,5,5,5,6])

        #content is formatted into a table
        a=table.draw()
        print('\n\n\n\n\n')
        print(a)

        #disconnects from the server
        db.close()


#--------------------------------------FUNCTION 6-------------------------
-------------

def choice2():
    '''This function will display the whole game history'''

    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor = db.cursor()

    #executing SQL query to display all the data from the table
    cursor.execute('SELECT * FROM playerinfo')

    #fetching the data using fetchall() method
    data=cursor.fetchall()
    #counting the number of rows
    row=cursor.rowcount

    #prints 'Sorry there are no entries' if there are no rows found
    if row == 0:
        print('Sorry, There are no entries')

    #displays the data if there are rows found
    else:

        #organizing the data in table format using the texttable module
        table=texttable.Texttable()

        #Creating the heading names for the table
        headings=['ID','Date','Time','Name','Mode','Word','Turns','TurnsUs
ed','Hint','Status']
        table.header(headings)


        #inserts the rows of data into the table
        for row in data:
            table.add_row(row)

        #adjusting column width
        table.set_cols_width([5,15,7,15,8,15,5,5,5,6])

        #content is formatted into a table
        a=table.draw()
        print('\n\n\n\n\n')
        print(a)

        #disconnects from the server
        db.close()


#--------------------------------------FUNCTION 7------------------------
-------------

def choice3():
    '''This function will display the game history of the selected
player'''

    #getting the name of the player from the user
    name=input("Enter the player's name to check the game history:
").upper()
```

```
    #opens the database
    db=game_hist.open_db()

    #prepare a cursor object using cursor() method
    cursor=db.cursor()

    #executing SQL query to display the data from the table with a
condition
    cursor.execute("SELECT * FROM playerinfo WHERE name = '"+str(name)+"'
")

    #fetching the data using fetchall() method
    data=cursor.fetchall()

    #counting the number of rows
    row=cursor.rowcount

    #prints 'Sorry there are no entries' if there are no rows found
    if row == 0:
        print('Sorry, There are no entries')

    #displays the data if there are rows found
    else:

        #organizing the data in table format using the texttable module
        table=texttable.Texttable()

        #Creating the heading names for the table
        headings=['ID','Date','Time','Name','Mode','Word','Turns','TurnsUs
ed','Hint','Status']
        table.header(headings)

         #inserts the rows of data into the table
        for row in data:
            table.add_row(row)

        #adjusting column width
        table.set_cols_width([5,15,7,15,8,15,5,5,5,6])

        #content is formatted into a table
        a=table.draw()
        print('\n\n\n\n\n')
        print(a)
        #disconnects from the server
        db.close()


#--------------------------------------FUNCTION 8-------------------------
-------------

def view_hist():
    '''This function will display a mini menu for the user to select from
which data the user needs to view'''

    while True:

        #assigning the variables(empty)
        c=''
        choice=0

        #exception handling
        try:

            #displays the three options for the user to choose from
            print('\n\n')
            print('1. I want to view the game history for this session')
            print('2. I want to view the whole game history')
            print('3. I want to view the game history of a specific
player')
            print()

            #asks the user for the choice
            c=input('Enter your choice: ')

            common.Exit(c)
            choice=int(c)
            common.clearConsole()

        except ValueError:
            #if anything else other than "EXIT" or an integer is given as
input, an error message will be displayed
            print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
            common.clearConsole()
            continue

        else:
            #executes the functions for choice1
            if choice==1:
```

```
    game_hist.choice1()
    break
#executes the functions for choice2
elif choice==2:
    game_hist.choice2()
    break
#executes the functions for choice3
elif choice==3:
    game_hist.choice3()
    break
#displays an error message if a wrong integer is entered
else:
    print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
    common.clearConsole()
    continue
```

3.2.4 game_rules.py

This module contains 1 function that will function if the user wishes to view the rules.

- Rules()

  This function prints the rules of the hangman game

<u>Code for game_rules.py</u>

```
def rules():
    '''This function prints the rules of the hangman game'''

    print()
    print()
    print('\u25CF  Hangman is a quick and easy single player game\n')
    print('\u25CF  The player is presented with some empty spaces to guess 
a word\n')
    print('\u25CF  There are 3 modes each player can choose from; Easy, 
Medium and Hard\n')
    print('\u25CF  The player guesses the word by entering a letter the 
word contains')
    print('   All places of that letter will be revealed, if the guess is 
correct')
    print('   However, every wrong guess brings the player one step close 
to losing\n')
    print('\u25CF  The player will have limited number of guesses')
    print('   The initial number of turns(guesses) is equal to the number 
of characters in the word\n')
    print('\u25CF  The player will be able to obtain one hint/clue of the 
word\n')
    print('\u25CF  The player wins the game by guessing the correct word 
within the given turns\n')
    print('\u25CF  The game will end if the player guesses the correct 
word, or if the player runs out of turns\n')
```

3.2.5 game_stats

This module contains 2 functions that will function if the user wishes to view the game stats.

The functions are as follows:

- stats()

  This function will get and return the Total games played, Total losses and wins

- html()

  This function will display the stat data in an html file

Code for game_stats.py

```
#---------------------------------IMPORT MODULES-------------------------------------------
--------
#user modules
import modules.game_hist as game_hist
import modules.common as common


#----------------------------------DEFINING FUNCTIONS----------------------------------------
-------------
#--------------------------------------FUNCTION 1-----------------------------------------
-------------

def stats():
    '''
    This function will get the Total games played, Total losses and wins

    Returns:
        TotalGames: This contains the value of the total number of games 
played
        TotalWins: This contains the value of the total number of wins
        TotalLoss: This contains the total number of losses
    '''

    while True:

        #Assigning variable(empty)
        choice=0
        c=0

        #exception handling
        try :
            #displaying a mini menu system to get the choice from the user
            print('\n\n')
            print('I want to view the stats for:')
            print('1. The whole game history')
            print('2. This session')
            print()

            #getting the choice from the user
            c=input('Enter your choice: ')
            common.Exit(c)
            choice=int(c)
            common.clearConsole()

        #raises an error if a value other than an integer is input
        except ValueError:
            print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
            common.clearConsole()
            continue
        else:
            #executes the function for choice1
            if choice==1:

                #opens the database
                db=game_hist.open_db()

                #prepare a cursor object using cursor() method
                cursor=db.cursor()

                #executing SQL query to display all the data from the
table

                cursor.execute('SELECT * FROM playerinfo')

                data=cursor.fetchall()
                TotalGames=cursor.rowcount

                #executing SQL query to display all the data from the
table according to a condition
                cursor.execute('SELECT * FROM playerinfo WHERE
winORloss="WIN"')
```

```
                data=cursor.fetchall()
                TotalWins=cursor.rowcount

                #executing SQL query to display all the data from the
table according to a condition
                cursor.execute('SELECT * FROM playerinfo WHERE
winORloss="LOSS"')

                data=cursor.fetchall()
                TotalLoss=cursor.rowcount

                break

            elif choice==2:
                #executes the functions for choice2
                db=game_hist.open_db()
                cursor=db.cursor()

                #executing SQL query to display all the data from the
table
                cursor.execute('SELECT * FROM playerinfo_persession')

                data=cursor.fetchall()
                TotalGames=cursor.rowcount

                #executing SQL query to display all the data from the
table according to a condition
                cursor.execute('SELECT * FROM playerinfo_persession WHERE
winORloss="WIN"')

                data=cursor.fetchall()
                TotalWins=cursor.rowcount

                #executing SQL query to display all the data from the
table according to a condition
                cursor.execute('SELECT * FROM playerinfo_persession WHERE
winORloss="LOSS"')

                data=cursor.fetchall()
                TotalLoss=cursor.rowcount

                break

            #displays an error message if a wrong integer is entered
            else:
                print('\nINVALID INPUT!!\nPlease enter a valid value!!\n')
                common.clearConsole()
                continue


    return TotalGames,TotalWins,TotalLoss


#-------------------------------------FUNCTION 2-------------------------
-------------

def html():
    '''This function will display the stat data in an html file'''

    #importing built in modules
    import webbrowser

    #getting the variable values from the function
    TotalGames,TotalWins,TotalLoss=stats()

    #opening an html file in write mode
    f=open('game_stats.html','w')

    #assigning the html code to a variable
    html_code=f'''<html>
    <head>
    <title>Hangman game stats</title>
    </head>
    </br>
    <body style="background-color:#0D306E;">

    <h1 style="background-color:#6D7C95;"><center><font face="Arial"
color="White" size="10">Hangman stats
    </font><center></h1>


    </br>
    <h2><font face="Arial" color="White" size="5">Total Number of
games:    %s </font></h2>
    <h2><font face="Arial" color="White" size="5">Total Number of
wins:     %s</font></h2>
    <h2><font face="Arial" color="White" size="5">Total Number of
losses:   %s</font></h2>
```

```
</body>
</html>'''

#inserting the python variables to the html code
final=html_code % (TotalGames,TotalWins,TotalLoss)

#writing the html code into the html file
f.write(final)

#closing the html file
f.close()


#displaying the stats
print('\t---------------------------------------')
print('\t     STATS FOR THE HANGMAN GAME')
print('\t---------------------------------------')

print('\t|Total Games played :    ',TotalGames)
print('\t|Total Wins:             ',TotalWins)
print('\t|Total Losses:           ',TotalLoss)

#opening the html file which contains the stats
webbrowser.open('game_stats.html')
```

3.3 Built-in modules

Few built in modules were used in the program to improve the quality of the program. The modules used in the program are as follows:

- random

  Python Random module is an in-built module of Python which is used to generate random numbers or words. This was used to generate a random number and get a random word for the hangman game

- datetime

  Python Datetime module supplies classes to work with date and time. This was used to get the current date when the program is running

- time

  The Python time module provides many ways of representing time in code. This was used to get the current time when the program is running

- os

  The OS module in Python provides functions for interacting with the operating system. This was used to give commands to clear the console

- webbrowser

  In Python, webbrowser module is a convenient web browser controller. This was used to open the html document in the web browser.

3.4 External Packages

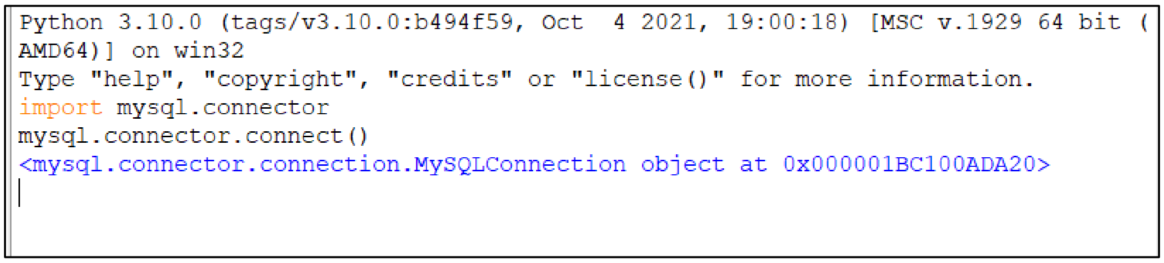
Few external packages were also used in the program to develop the program. The following are the packages used and their respective ways to install them:

- Mysql.connector

  MySQL Connector enables Python programs to access MySQL databases.

  It is connected by running the following code in python –

  mysql.connector.connect()



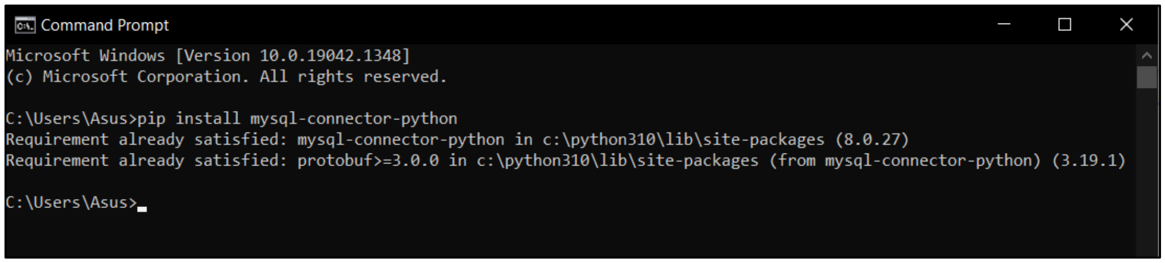
```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
import mysql.connector
mysql.connector.connect()
<mysql.connector.connection.MySQLConnection object at 0x000001BC100ADA20>
```

Figure 2(3.4.1): mysqlconnection

To install the mysql connector the following command should be executed in the terminal –

pip install mysql-connector-python

If it is already installed, you will see the following output in the terminal:



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in c:\python310\lib\site-packages (8.0.27)
Requirement already satisfied: protobuf>=3.0.0 in c:\python310\lib\site-packages (from mysql-connector-python) (3.19.1)

C:\Users\Asus>
```

Figure 3(3.4.2):mysql.connector.connect installation

More instructions on how to install mysql.connector:

https://www.a2hosting.com/kb/developer-corner/mysql/connecting-to-mysql-using-python

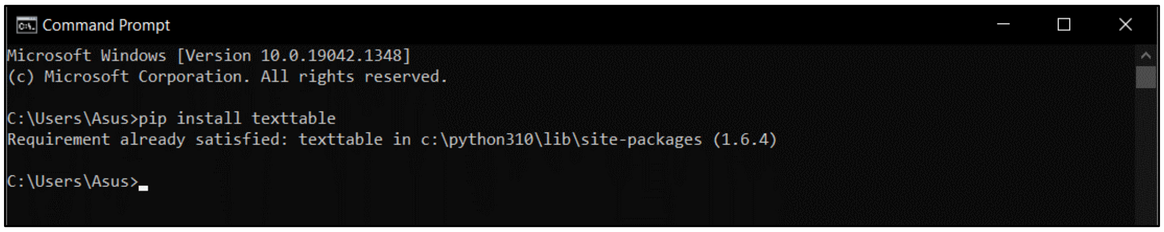
- Texttable

  It is a python module, which helps us to print table on terminal

  To install the texttable package the following command should be executed in the terminal –

  pip install texttable

  If it is already installed, you will see the following output in the terminal:



Command Prompt

```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>pip install texttable
Requirement already satisfied: texttable in c:\python310\lib\site-packages (1.6.4)

C:\Users\Asus>
```

Figure 4(3.4.3): texttable installation

More instructions on how to install texttable package:
https://www.geeksforgeeks.org/texttable-module-in-python/

3.5 MySQL Database

The databases were used to keep records of the full game history as well as the game history per session. This was done in the phpMyAdmin through the xampp server. Following are the details of the database used:

username – *'root'*

password – No password

server – '*localhost'*

database – '*hangman'*

Two tables, namely, playerinfo and playerinfo_persession were created. The two tables have been exported as hangman.sql. The databases must be connected to run the hangman program.

<u>Steps to import the data tables:</u>

1. In the MySQL server, create a database named 'hangman' by clicking on new

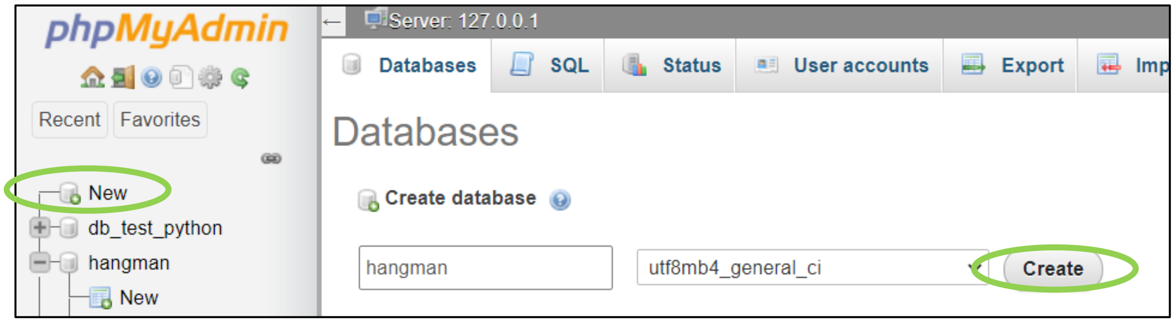


Figure 5(3.5.1):Import database

2. Once the database is created, select that database, and go to 'import'

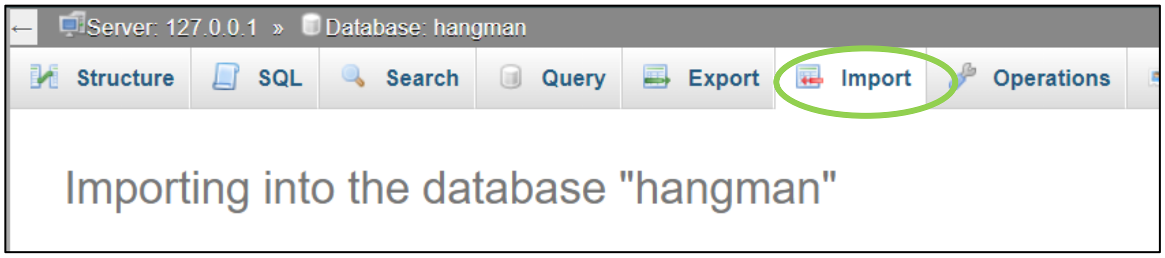
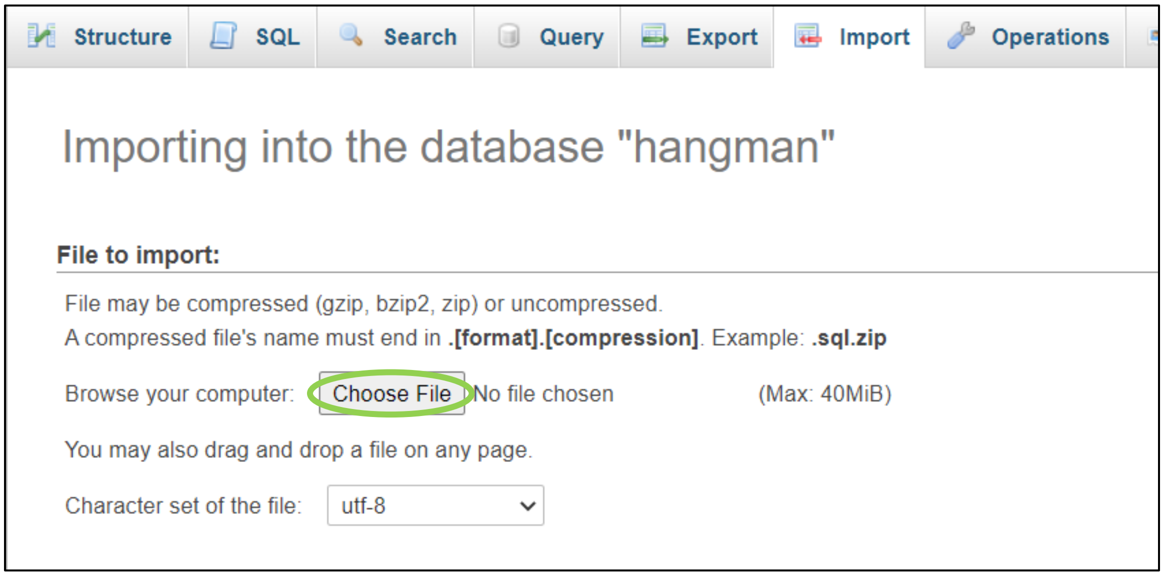

Server: 127.0.0.1 » Database: hangman
Structure
SQL
Search
Query
Export
Import
Operations
Importing into the database "hangman"

Figure 6(3.5.2):Import database1

3. Under ‘File to import:’ click on the ‘choose file’



Importing into the database "hangman"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

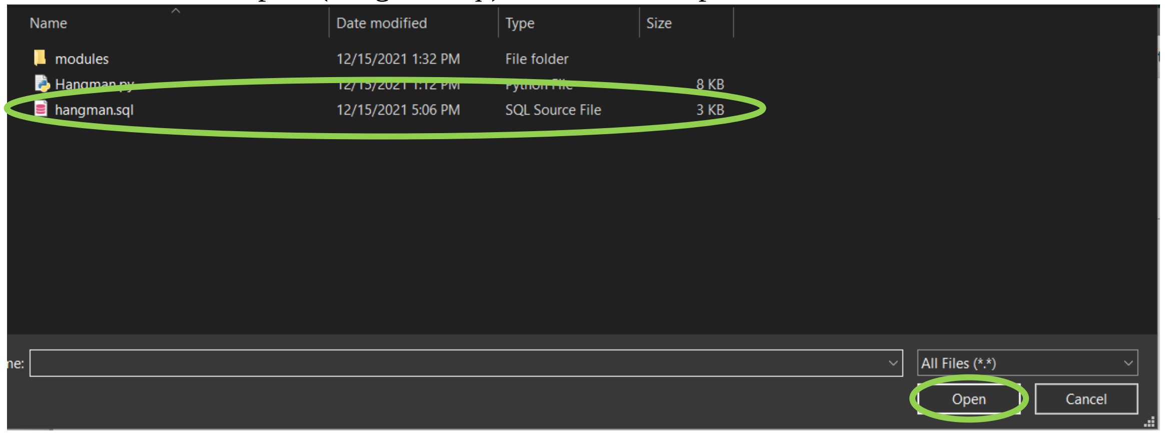
Browse your computer: Choose File No file chosen (Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Figure 7(3.5.3):Import database2

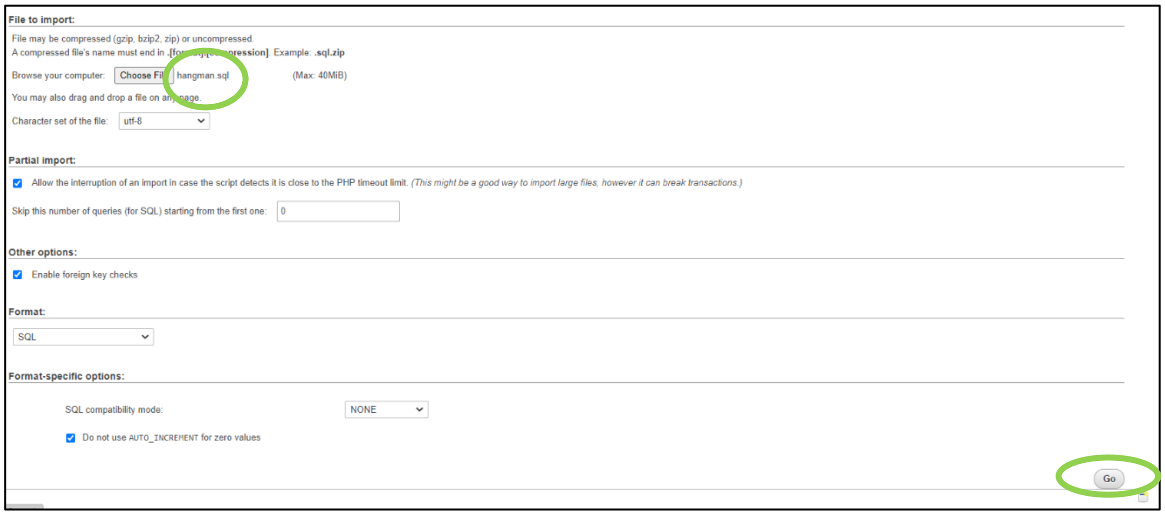
4. Select the file to import (hangman.sql), and click on open



Name	Date modified	Type	Size
modules	12/15/2021 1:32 PM	File folder	
Hangman.py	12/15/2021 1:12 PM	Python File	8 KB
hangman.sql	12/15/2021 5:06 PM	SQL Source File	3 KB

Figure 8(3.5.4):Import database3

5. Once the file is chosen click on ‘go’, at the left bottom



File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[fo[illegible]].[c[illegible]pression]**. Example: **.sql.zip**

Browse your computer: Choose Fi[illegible] hangman.sql (Max: 40MiB)

You may also drag and drop a file on an[illegible] page.

Character set of the file: utf-8

Partial import:

✓ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) starting from the first one: 0

Other options:

✓ Enable foreign key checks

Format:

SQL

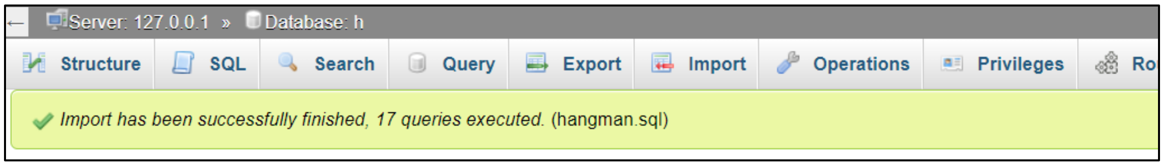
Format-specific options:

SQL compatibility mode: NONE

✓ Do not use `AUTO_INCREMENT` for zero values

Figure 9(3.5.5):Import database4

6. If the import was done successfully, the following will appear



✓ *Import has been successfully finished, 17 queries executed.* (hangman.sql)

Figure 10(3.5.6):Import database5

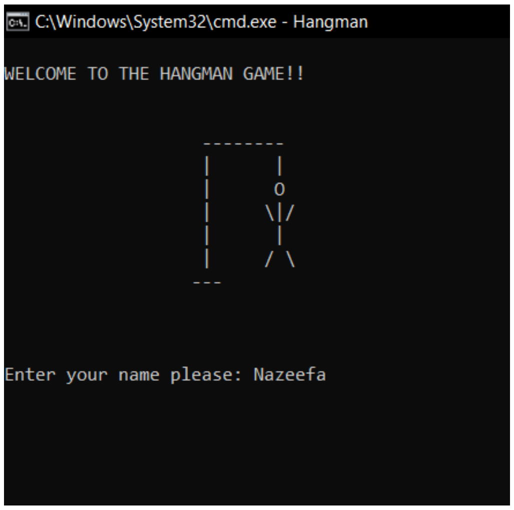
3.6 Html

An html document will be created when the program runs. This html document will display the stats such as the total games played, total wins and total losses of both the whole game history and for per session of the hangman game.

4. Screenshots of the running program

Note: Once an input is entered to the console program, it has been programmed to clear the console to improve the user experience.

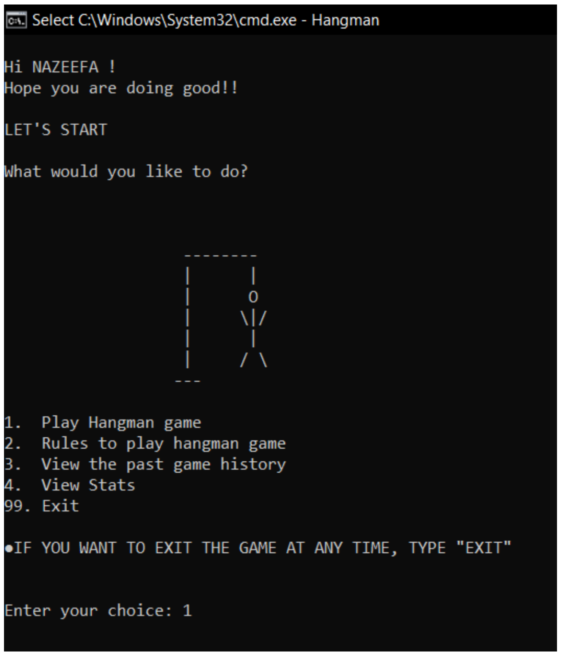
1. Welcomes the user and gets the user's name



```
C:\Windows\System32\cmd.exe - Hangman

WELCOME TO THE HANGMAN GAME!!


                  --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---



Enter your name please: Nazeefa
```

Figure 11(4.1):Welcomes the user

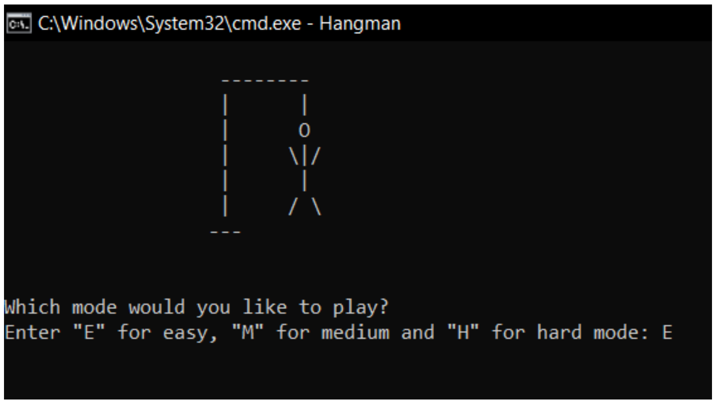
2. Main menu - Gets the choice from the user



```
Select C:\Windows\System32\cmd.exe - Hangman

Hi NAZEEFA !
Hope you are doing good!!

LET'S START

What would you like to do?



                  --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---

1.  Play Hangman game
2.  Rules to play hangman game
3.  View the past game history
4.  View Stats
99. Exit

•IF YOU WANT TO EXIT THE GAME AT ANY TIME, TYPE "EXIT"


Enter your choice: 1
```

Figure 12(4.2): main menu

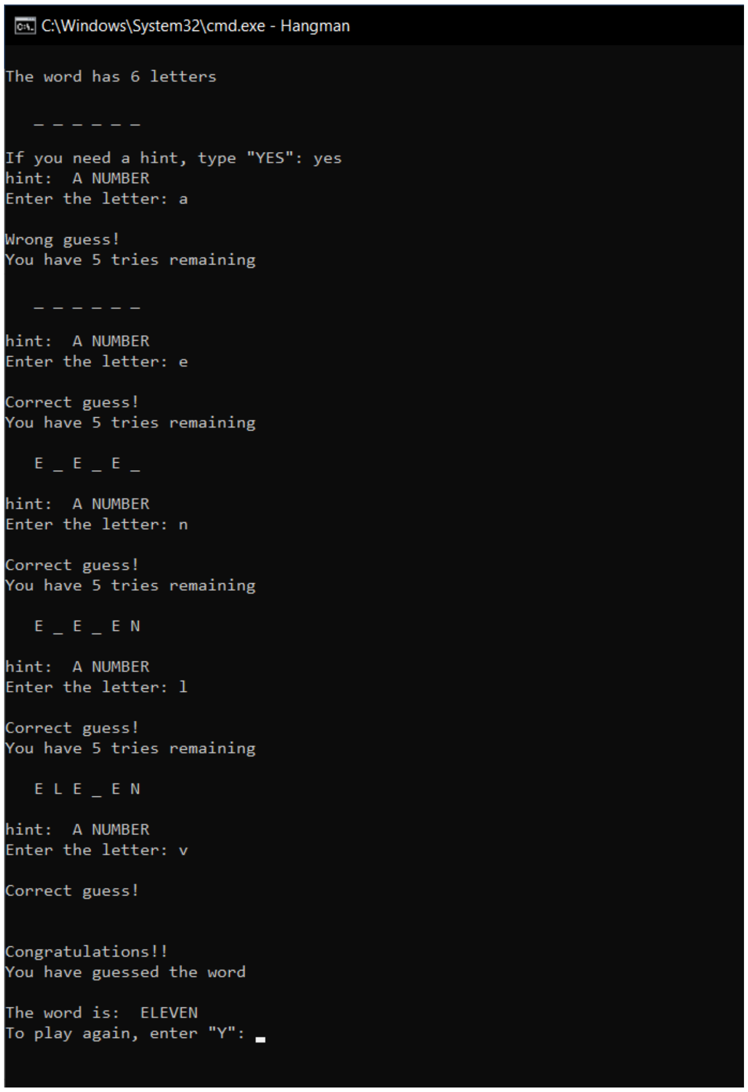
3. Game mode (Menu – choice 1)


C:\Windows\System32\cmd.exe - Hangman

```
                   --------
                    |      |
                    |      O
                    |     \|/
                    |      |
                    |     / \
                   ---


Which mode would you like to play?
Enter "E" for easy, "M" for medium and "H" for hard mode: E
```

Figure 13(4.3):Game mode

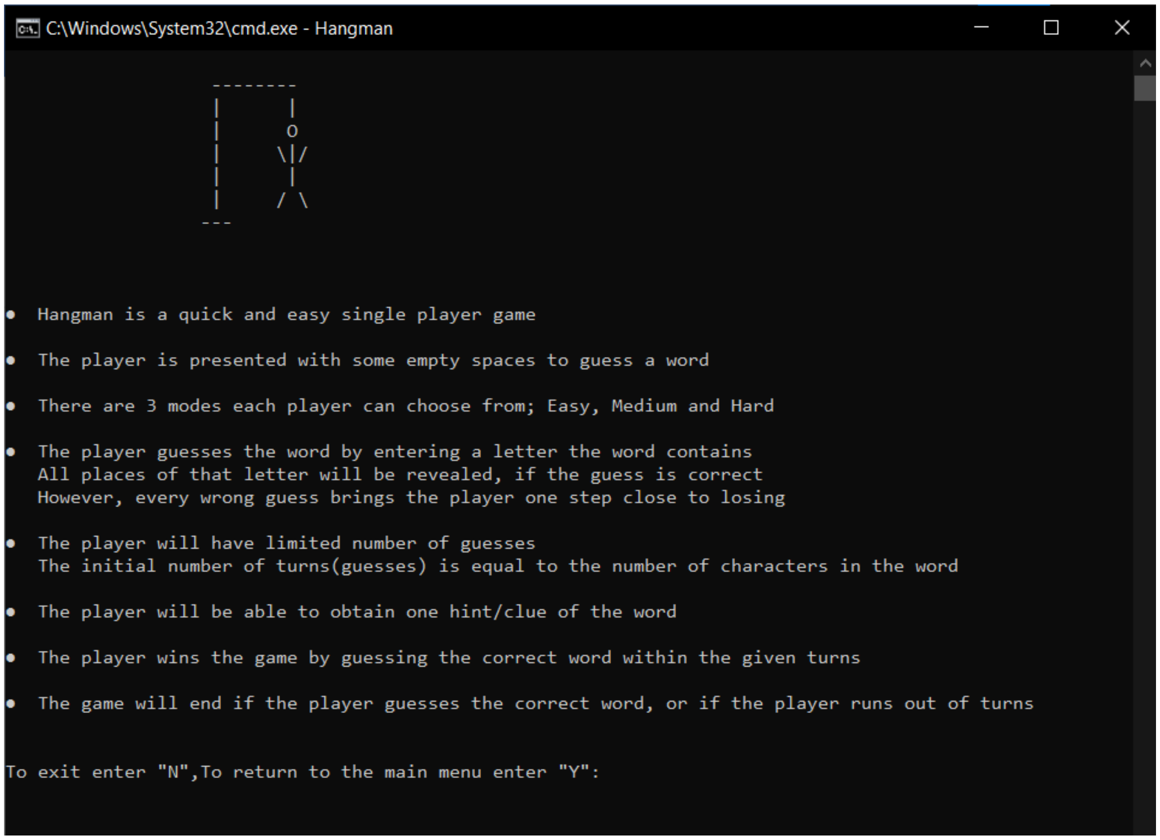
4. Game


C:\Windows\System32\cmd.exe - Hangman

```
The word has 6 letters

   _ _ _ _ _ _

If you need a hint, type "YES": yes
hint:  A NUMBER
Enter the letter: a

Wrong guess!
You have 5 tries remaining

   _ _ _ _ _ _

hint:  A NUMBER
Enter the letter: e

Correct guess!
You have 5 tries remaining

   E _ E _ E _

hint:  A NUMBER
Enter the letter: n

Correct guess!
You have 5 tries remaining

   E _ E _ E N

hint:  A NUMBER
Enter the letter: l

Correct guess!
You have 5 tries remaining

   E L E _ E N

hint:  A NUMBER
Enter the letter: v

Correct guess!


Congratulations!!
You have guessed the word

The word is:  ELEVEN
To play again, enter "Y":
```

Figure 14(4.4):Game

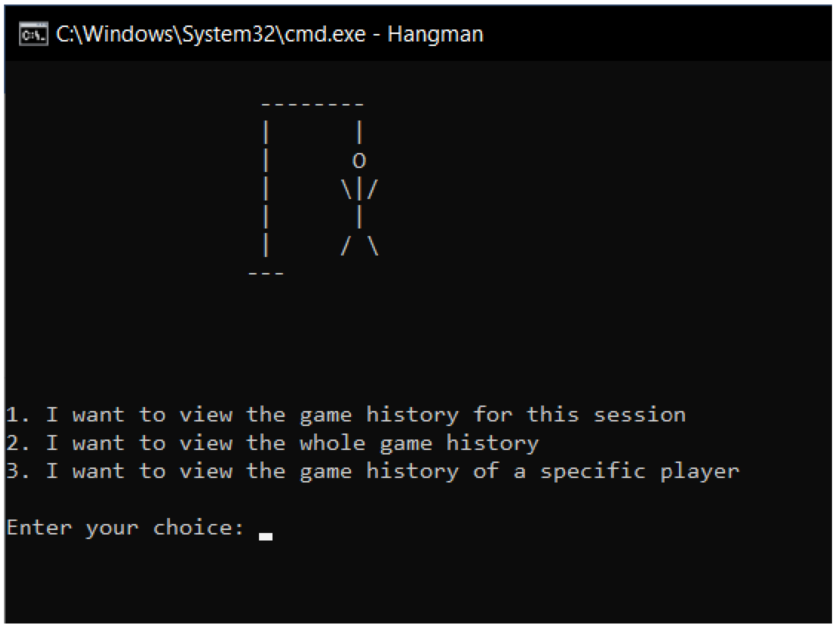
5. Rules (Menu – choice 2)



```
C:\Windows\System32\cmd.exe - Hangman

                   --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---



•  Hangman is a quick and easy single player game

•  The player is presented with some empty spaces to guess a word

•  There are 3 modes each player can choose from; Easy, Medium and Hard

•  The player guesses the word by entering a letter the word contains
   All places of that letter will be revealed, if the guess is correct
   However, every wrong guess brings the player one step close to losing

•  The player will have limited number of guesses
   The initial number of turns(guesses) is equal to the number of characters in the word

•  The player will be able to obtain one hint/clue of the word

•  The player wins the game by guessing the correct word within the given turns

•  The game will end if the player guesses the correct word, or if the player runs out of turns


To exit enter "N",To return to the main menu enter "Y":
```

Figure 15(4.5):Rules

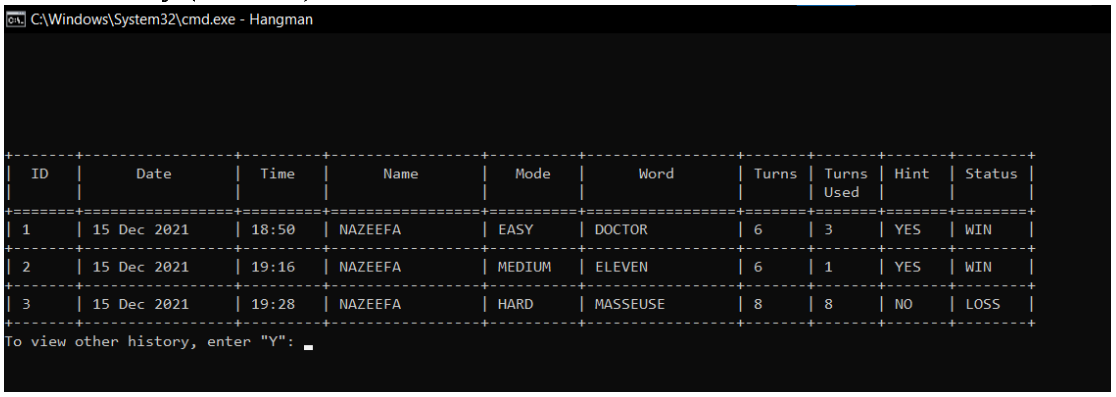
6. Game history (Menu – choice 3)



```
C:\Windows\System32\cmd.exe - Hangman

                   --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---



1. I want to view the game history for this session
2. I want to view the whole game history
3. I want to view the game history of a specific player

Enter your choice: _
```

Figure 16(4.6): Game history

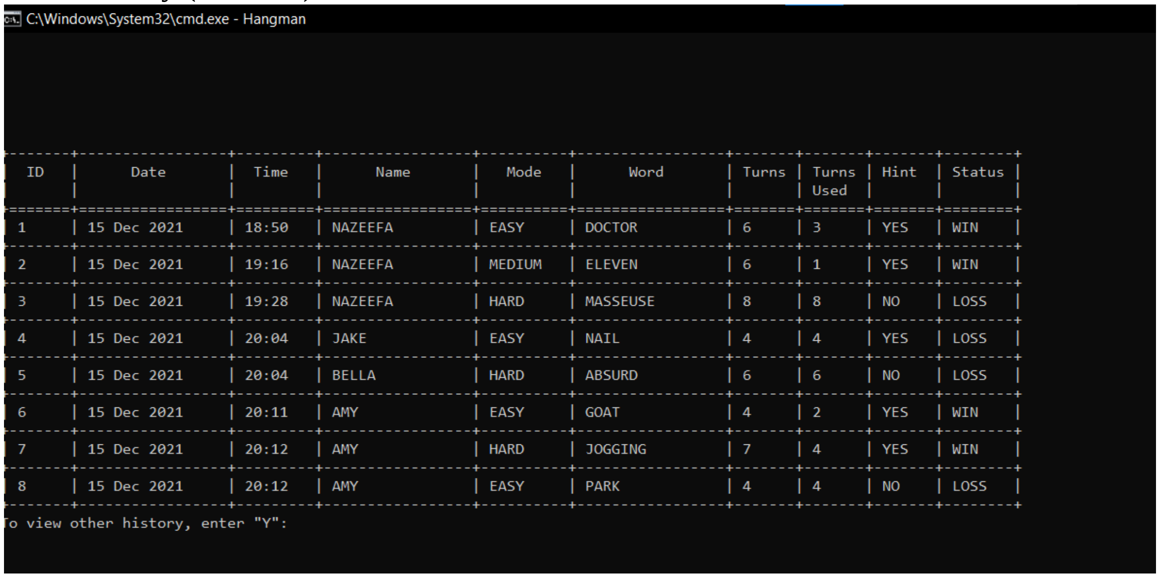
7. Game history (choice 1)



```
C:\Windows\System32\cmd.exe - Hangman

+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
|  ID   |      Date      |  Time   |      Name      |   Mode   |      Word      | Turns | Turns | Hint  | Status |
|       |                |         |                |          |                |       | Used  |       |        |
+=======+================+=========+================+==========+================+=======+=======+=======+========+
| 1     | 15 Dec 2021    | 18:50   | NAZEEFA        | EASY     | DOCTOR         | 6     | 3     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 2     | 15 Dec 2021    | 19:16   | NAZEEFA        | MEDIUM   | ELEVEN         | 6     | 1     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 3     | 15 Dec 2021    | 19:28   | NAZEEFA        | HARD     | MASSEUSE       | 8     | 8     | NO    | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
To view other history, enter "Y":
```

Figure 17(4.7):Game history1

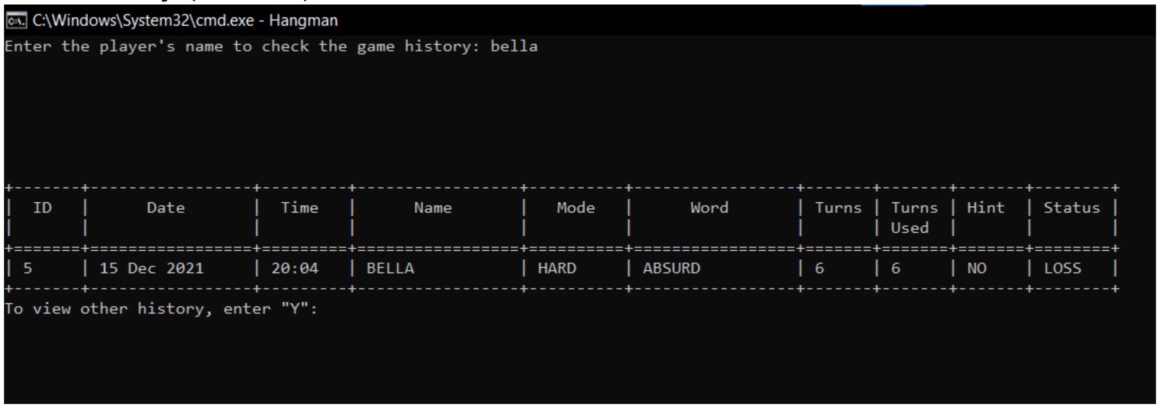
8. Game history (choice 2)



```
C:\Windows\System32\cmd.exe - Hangman

+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
|  ID   |      Date      |  Time   |      Name      |   Mode   |      Word      | Turns | Turns | Hint  | Status |
|       |                |         |                |          |                |       | Used  |       |        |
+=======+================+=========+================+==========+================+=======+=======+=======+========+
| 1     | 15 Dec 2021    | 18:50   | NAZEEFA        | EASY     | DOCTOR         | 6     | 3     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 2     | 15 Dec 2021    | 19:16   | NAZEEFA        | MEDIUM   | ELEVEN         | 6     | 1     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 3     | 15 Dec 2021    | 19:28   | NAZEEFA        | HARD     | MASSEUSE       | 8     | 8     | NO    | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 4     | 15 Dec 2021    | 20:04   | JAKE           | EASY     | NAIL           | 4     | 4     | YES   | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 5     | 15 Dec 2021    | 20:04   | BELLA          | HARD     | ABSURD         | 6     | 6     | NO    | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 6     | 15 Dec 2021    | 20:11   | AMY            | EASY     | GOAT           | 4     | 2     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 7     | 15 Dec 2021    | 20:12   | AMY            | HARD     | JOGGING        | 7     | 4     | YES   | WIN    |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
| 8     | 15 Dec 2021    | 20:12   | AMY            | EASY     | PARK           | 4     | 4     | NO    | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
To view other history, enter "Y":
```

Figure 18(4.8):Game history2

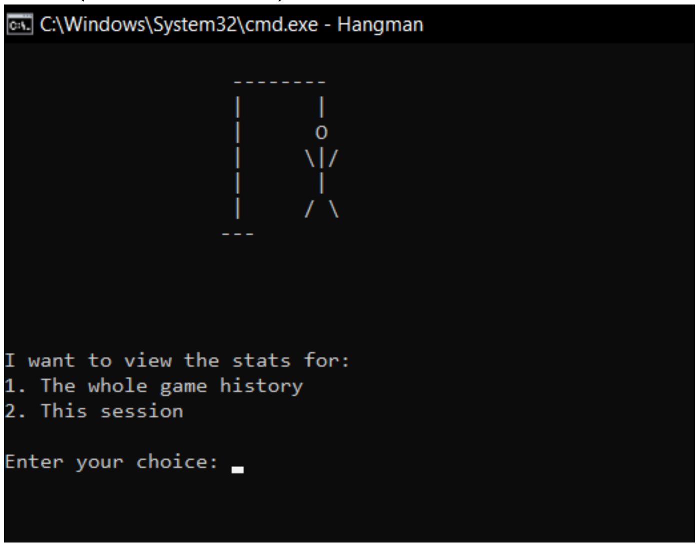
9. Game history (choice 3)



```
C:\Windows\System32\cmd.exe - Hangman
Enter the player's name to check the game history: bella

+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
|  ID   |      Date      |  Time   |      Name      |   Mode   |      Word      | Turns | Turns | Hint  | Status |
|       |                |         |                |          |                |       | Used  |       |        |
+=======+================+=========+================+==========+================+=======+=======+=======+========+
| 5     | 15 Dec 2021    | 20:04   | BELLA          | HARD     | ABSURD         | 6     | 6     | NO    | LOSS   |
+-------+----------------+---------+----------------+----------+----------------+-------+-------+-------+--------+
To view other history, enter "Y":
```

Figure 19(4.9):Game history3

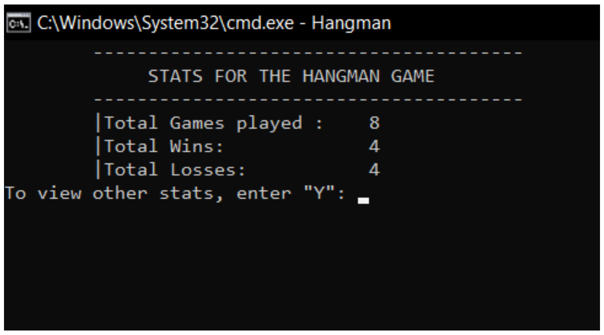
10. Stats (Menu-choice 4)



```
C:\Windows\System32\cmd.exe - Hangman

                   --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---



I want to view the stats for:
1. The whole game history
2. This session

Enter your choice:
```

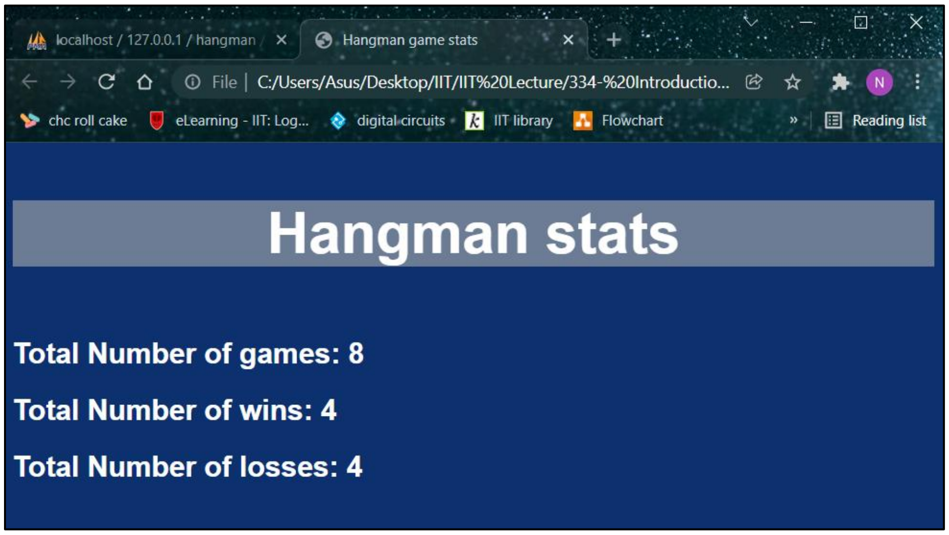
Figure 20(4.10):Stats

11. Stats (Choice 1)



```
C:\Windows\System32\cmd.exe - Hangman
        ----------------------------------------
             STATS FOR THE HANGMAN GAME
        ----------------------------------------
        |Total Games played :    8
        |Total Wins:             4
        |Total Losses:           4
To view other stats, enter "Y":
```

Figure 21(4.11):Stats1



Hangman stats

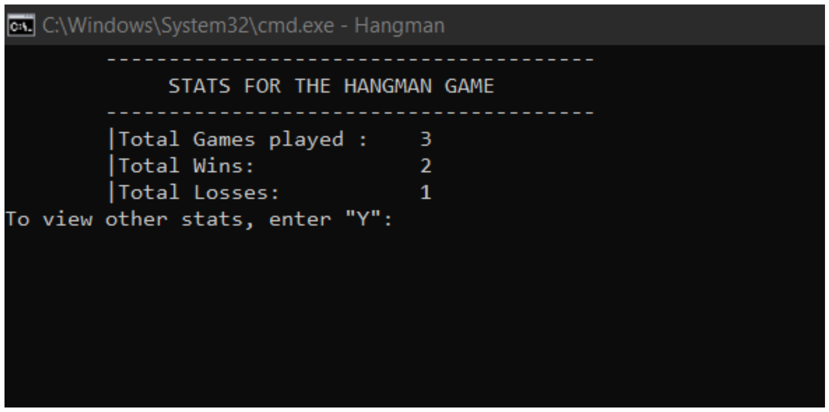
Total Number of games: 8

Total Number of wins: 4

Total Number of losses: 4

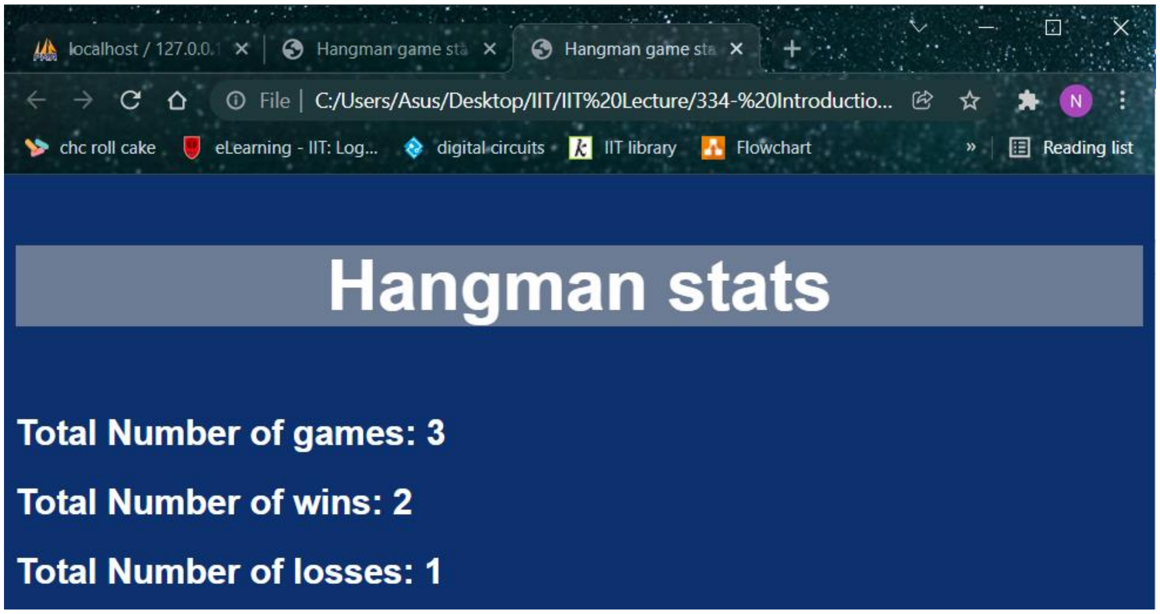
Figure 22(4.12):StatsHTML

12. Stats (choice 2)



```
        --------------------------------------
             STATS FOR THE HANGMAN GAME
        --------------------------------------
        |Total Games played :    3
        |Total Wins:             2
        |Total Losses:           1
To view other stats, enter "Y":
```

Figure 23(4.13):Stats2



Hangman stats

Total Number of games: 3

Total Number of wins: 2

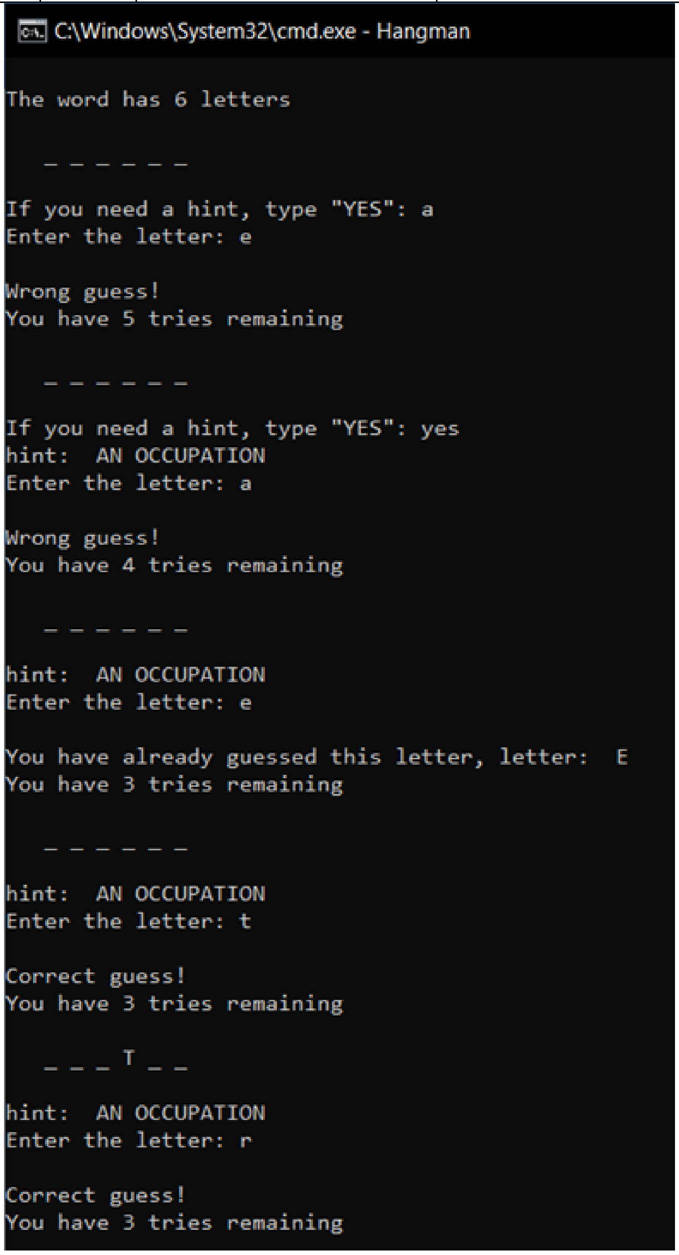
Total Number of losses: 1

Figure 24(4.14):statsHTML1

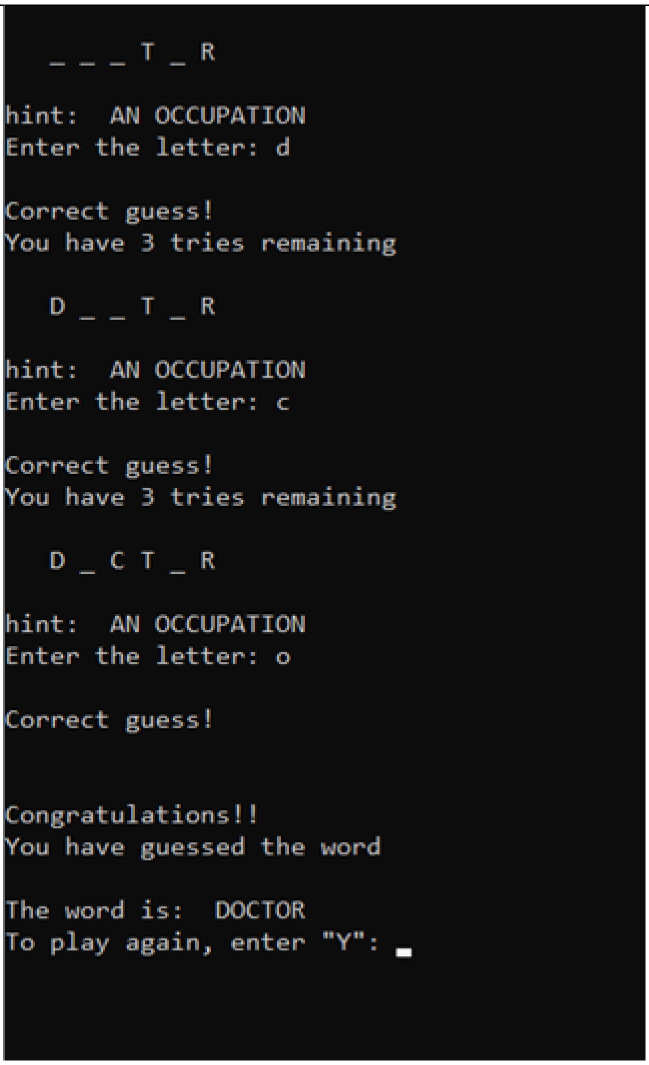
5. Test cases

5.1 Hangman game

Test case #	Word	Inputs			Expected output	Actual output	Remarks
		Mode (E/M/H)	Hint (YES)	letter			
1	doctor	E	a	e	Wrong guess! You have 5 tries remaining	Wrong guess! You have 5 tries remaining	Pass
		-	yes	a	hint: AN OCCUPATION Wrong guess! You have 4 tries remaining	hint: AN OCCUPATION Wrong guess! You have 4 tries remaining	Pass
		-	-	e	You have already guessed this letter, letter: E You have 3 tries remaining	You have already guessed this letter, letter: E You have 3 tries remaining	Pass
		-	-	t	Correct guess! You have 3 tries remaining	Correct guess! You have 3 tries remaining	Pass
		-	-	r	Correct guess! You have 3 tries remaining	Correct guess! You have 3 tries remaining	Pass
		-	-	d	Correct guess! You have 3 tries remaining	Correct guess! You have 3 tries remaining	Pass
		-	-	c	Correct guess! You have 3 tries remaining	Correct guess! You have 3 tries remaining	Pass

		-	-	o	Correct guess! Congratulations!! You have guessed the word The word is: DOCTOR To play again, enter "Y":	Correct guess! Congratulations!! You have guessed the word The word is: DOCTOR To play again, enter "Y":	Pass
		 Figure 25(5.1.2):Testcase1					

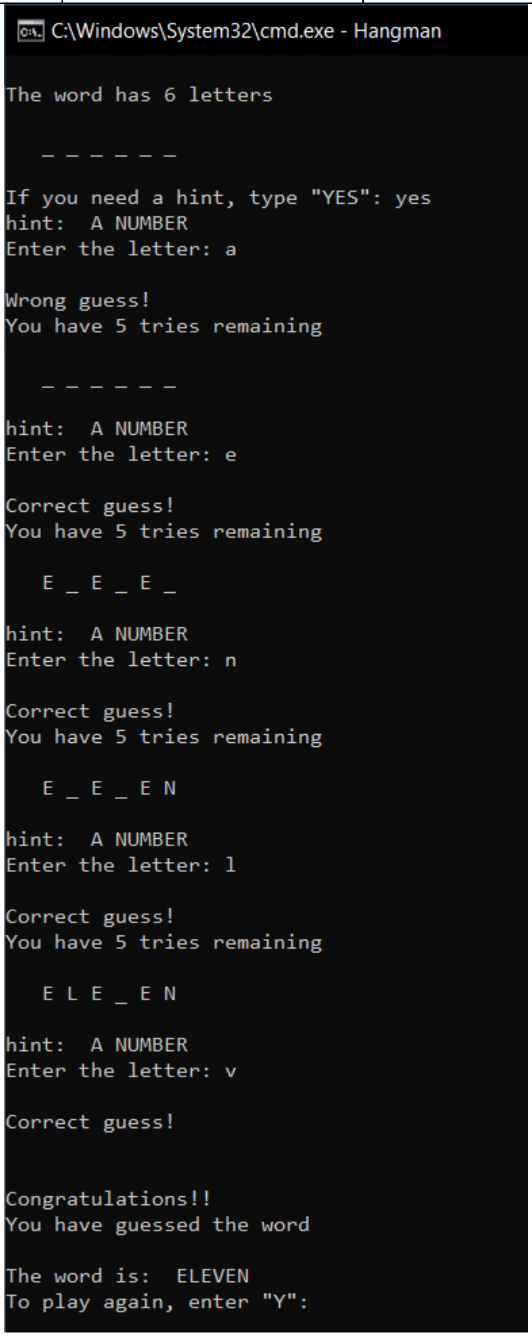
```
C:\Windows\System32\cmd.exe - Hangman

The word has 6 letters

   _ _ _ _ _ _

If you need a hint, type "YES": a
Enter the letter: e

Wrong guess!
You have 5 tries remaining

   _ _ _ _ _ _

If you need a hint, type "YES": yes
hint:  AN OCCUPATION
Enter the letter: a

Wrong guess!
You have 4 tries remaining

   _ _ _ _ _ _

hint:  AN OCCUPATION
Enter the letter: e

You have already guessed this letter, letter:  E
You have 3 tries remaining

   _ _ _ _ _ _

hint:  AN OCCUPATION
Enter the letter: t

Correct guess!
You have 3 tries remaining

   _ _ _ T _ _

hint:  AN OCCUPATION
Enter the letter: r

Correct guess!
You have 3 tries remaining
```



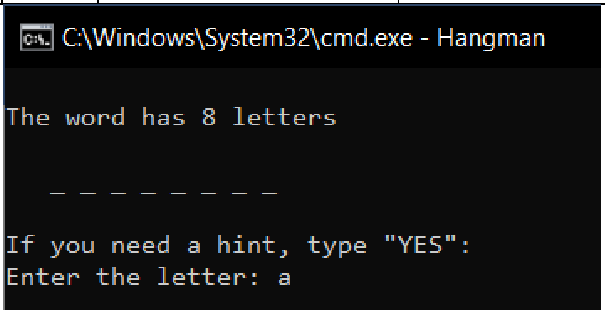
```
   _ _ _ T _ R

hint:  AN OCCUPATION
Enter the letter: d

Correct guess!
You have 3 tries remaining

   D _ _ T _ R

hint:  AN OCCUPATION
Enter the letter: c

Correct guess!
You have 3 tries remaining

   D _ C T _ R

hint:  AN OCCUPATION
Enter the letter: o

Correct guess!


Congratulations!!
You have guessed the word

The word is:  DOCTOR
To play again, enter "Y": _
```

Figure 26(5.1.3):Testcase2

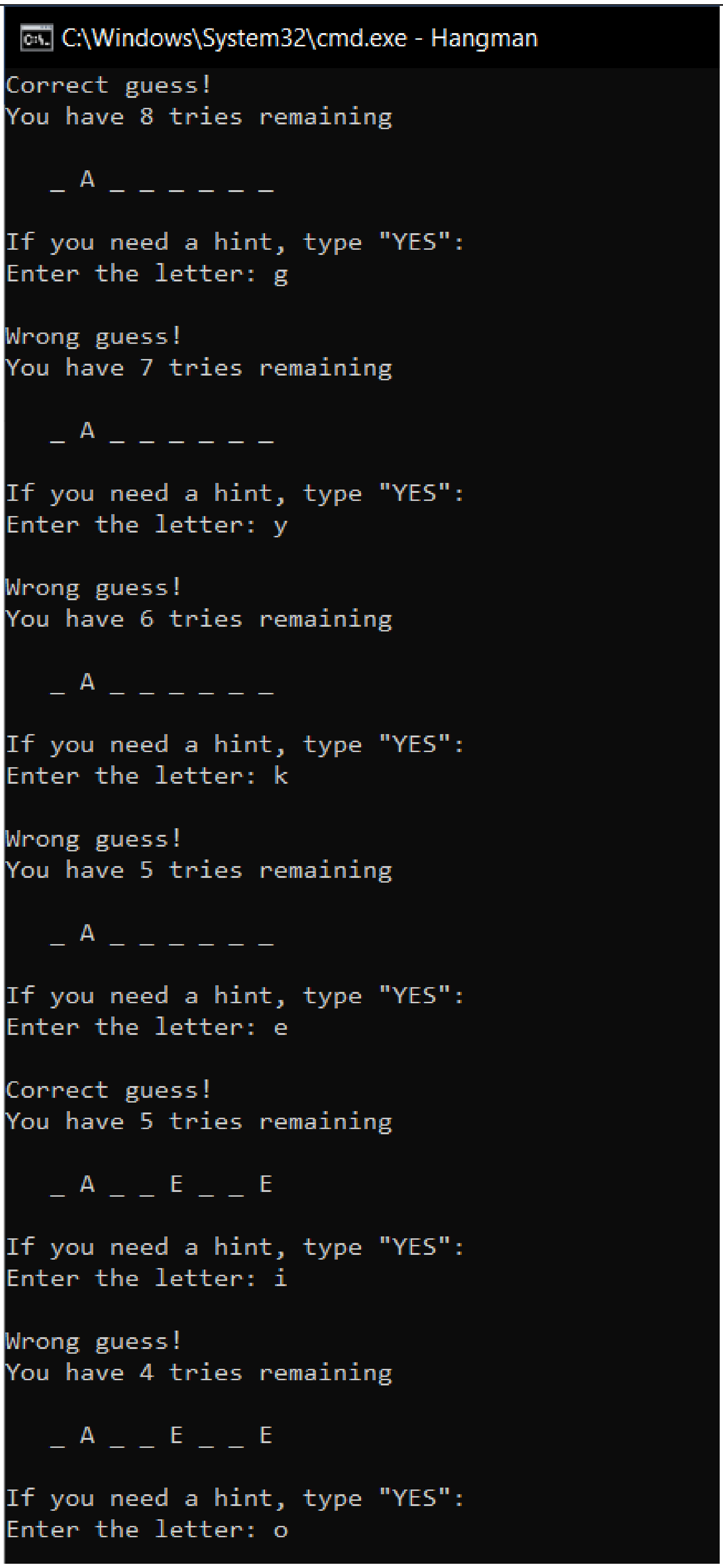
2 Retry	eleven	M	yes	a	hint: A NUMBER Wrong guess! You have 5 tries remaining	hint: A NUMBER Wrong guess! You have 5 tries remaining	pass
		-	-	e	Correct guess! You have 5 tries remaining	Correct guess! You have 5 tries remaining	Pass
		-	-	n	Correct guess! You have 5 tries remaining	Correct guess! You have 5 tries remaining	Pass
		-	-	l	Correct guess!	Correct guess!	Pass

					You have 5 tries remaining	You have 5 tries remaining	
		-	-	v	Correct guess! Congratulations!! You have guessed the word	Correct guess! Congratulations!! You have guessed the word	Pass
		 Figure 27(5.1.4):Testcase3					

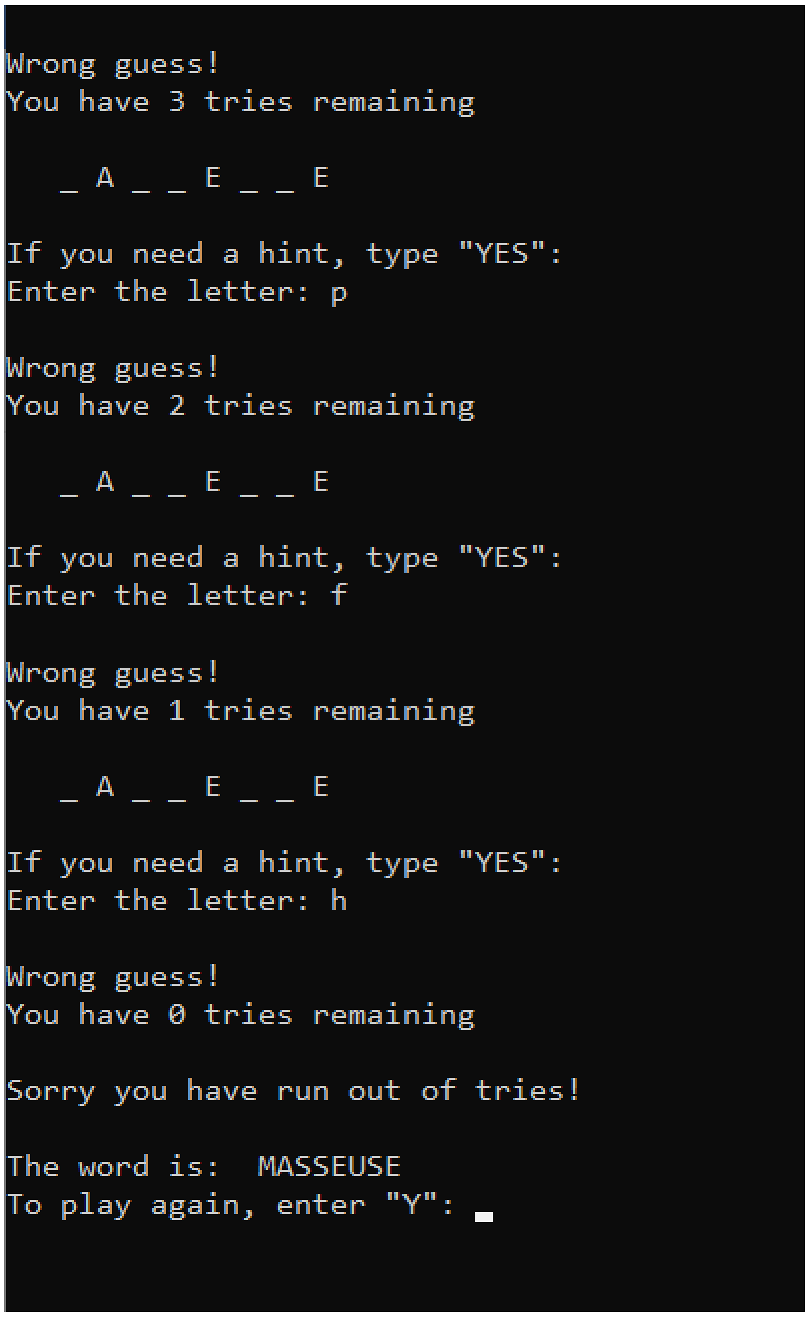
```
C:\Windows\System32\cmd.exe - Hangman

The word has 6 letters

   _ _ _ _ _ _

If you need a hint, type "YES": yes
hint:  A NUMBER
Enter the letter: a

Wrong guess!
You have 5 tries remaining

   _ _ _ _ _ _

hint:  A NUMBER
Enter the letter: e

Correct guess!
You have 5 tries remaining

   E _ E _ E _

hint:  A NUMBER
Enter the letter: n

Correct guess!
You have 5 tries remaining

   E _ E _ E N

hint:  A NUMBER
Enter the letter: l

Correct guess!
You have 5 tries remaining

   E L E _ E N

hint:  A NUMBER
Enter the letter: v

Correct guess!


Congratulations!!
You have guessed the word

The word is:  ELEVEN
To play again, enter "Y":
```

3 retry	masseuse	H	-	A	Correct guess! You have 8 tries remaining	Correct guess! You have 8 tries remaining	Pass
		-	-	G	Wrong guess! You have 7 tries remaining	Wrong guess! You have 7 tries remaining	Pass
		-	-	Y	Wrong guess! You have 6 tries remaining	Wrong guess! You have 6 tries remaining	Pass
		-	-	K	Wrong guess! You have 5 tries remaining	Wrong guess! You have 5 tries remaining	Pass
		-	-	E	Correct guess! You have 5 tries remaining	Correct guess! You have 5 tries remaining	Pass
		-	-	I	Wrong guess! You have 4 tries remaining	Wrong guess! You have 4 tries remaining	Pass
		-	-	O	Wrong guess! You have 3 tries remaining	Wrong guess! You have 3 tries remaining	Pass
		-	-	P	Wrong guess! You have 2 tries remaining	Wrong guess! You have 2 tries remaining	Pass
		-	-	F	Wrong guess! You have 1 tries remaining	Wrong guess! You have 1 tries remaining	Pass
		-	-	H	Wrong guess! You have 0 tries remaining Sorry you have run out of tries! The word is: MASSEUSE	Wrong guess! You have 0 tries remaining Sorry you have run out of tries! The word is: MASSEUSE	pass
		 *Figure 28(5.1.5):Testcase4*					

```
C:\Windows\System32\cmd.exe - Hangman

The word has 8 letters


  _ _ _ _ _ _ _ _

If you need a hint, type "YES":
Enter the letter: a
```

			 Figure 29(5.1.6):Testcase5

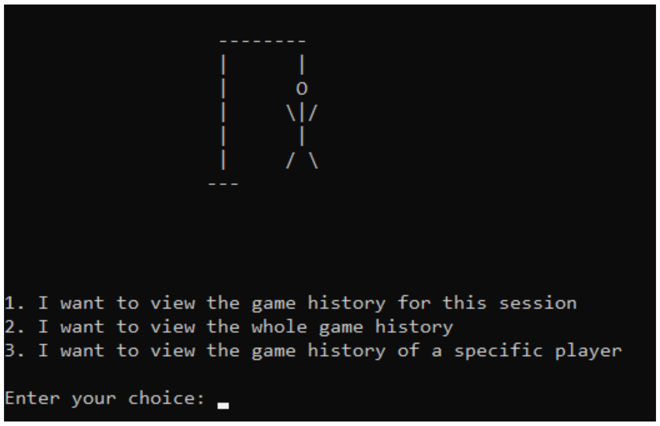
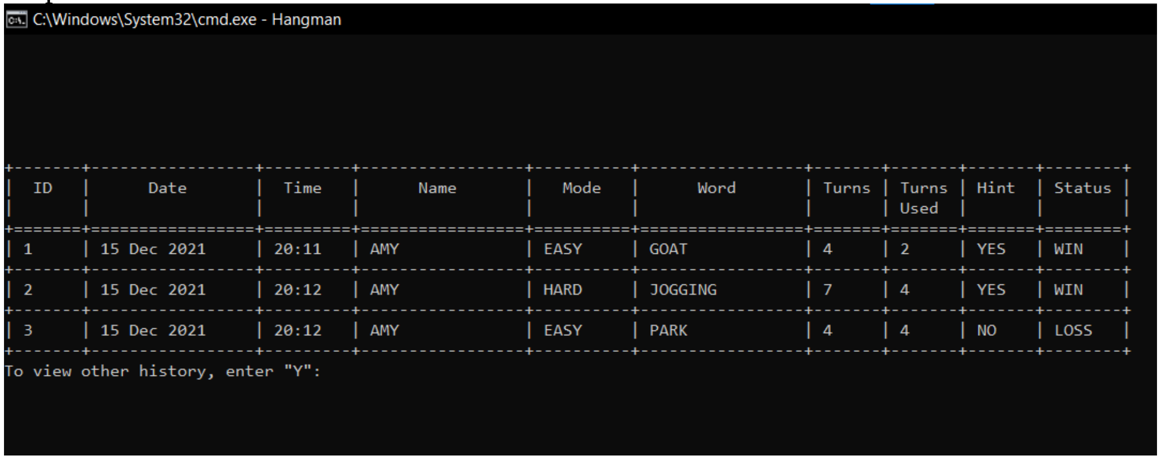
```
C:\Windows\System32\cmd.exe - Hangman
Correct guess!
You have 8 tries remaining

   _ A _ _ _ _ _ _

If you need a hint, type "YES":
Enter the letter: g

Wrong guess!
You have 7 tries remaining

   _ A _ _ _ _ _ _

If you need a hint, type "YES":
Enter the letter: y

Wrong guess!
You have 6 tries remaining

   _ A _ _ _ _ _ _

If you need a hint, type "YES":
Enter the letter: k

Wrong guess!
You have 5 tries remaining

   _ A _ _ _ _ _ _

If you need a hint, type "YES":
Enter the letter: e

Correct guess!
You have 5 tries remaining

   _ A _ _ E _ _ E

If you need a hint, type "YES":
Enter the letter: i

Wrong guess!
You have 4 tries remaining

   _ A _ _ E _ _ E

If you need a hint, type "YES":
Enter the letter: o
```

		 Figure 30(5.1.6):Testcase6

```
Wrong guess!
You have 3 tries remaining

   _ A _ _ E _ _ E

If you need a hint, type "YES":
Enter the letter: p

Wrong guess!
You have 2 tries remaining

   _ A _ _ E _ _ E

If you need a hint, type "YES":
Enter the letter: f

Wrong guess!
You have 1 tries remaining

   _ A _ _ E _ _ E

If you need a hint, type "YES":
Enter the letter: h

Wrong guess!
You have 0 tries remaining

Sorry you have run out of tries!

The word is:  MASSEUSE
To play again, enter "Y":
```

Table 1(5.1.1):Hangman game testcase

5.2 Viewing the game history

Test case #	Input Choice for the history	Expected output	Actual output	Remarks
				
1	1	(Table displaying 3 records)	(Table displaying 3 records)	Pass
	output: 			

```
 --------
 |      |
 |      O
 |     \|/
 |      |
 |     / \
---



1. I want to view the game history for this session
2. I want to view the whole game history
3. I want to view the game history of a specific player

Enter your choice:
```

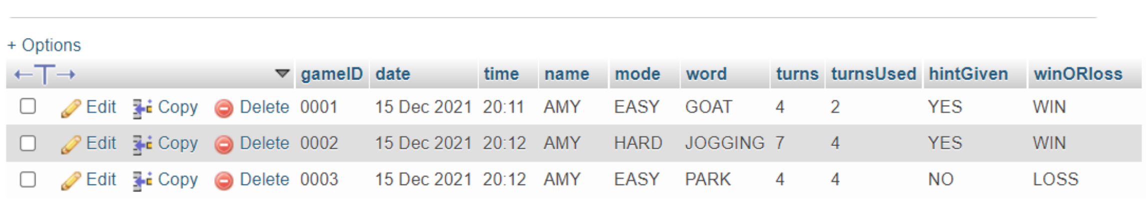
Figure 31(5.2.1):Testcase7

C:\Windows\System32\cmd.exe - Hangman

```
+-------+-----------------+---------+----------------+---------+-----------------+-------+-------+-------+--------+
|  ID   |      Date       |  Time   |      Name      |  Mode   |      Word       | Turns | Turns | Hint  | Status |
|       |                 |         |                |         |                 |       | Used  |       |        |
+=======+=================+=========+================+=========+=================+=======+=======+=======+========+
| 1     | 15 Dec 2021     | 20:11   | AMY            | EASY    | GOAT            | 4     | 2     | YES   | WIN    |
+-------+-----------------+---------+----------------+---------+-----------------+-------+-------+-------+--------+
| 2     | 15 Dec 2021     | 20:12   | AMY            | HARD    | JOGGING         | 7     | 4     | YES   | WIN    |
+-------+-----------------+---------+----------------+---------+-----------------+-------+-------+-------+--------+
| 3     | 15 Dec 2021     | 20:12   | AMY            | EASY    | PARK            | 4     | 4     | NO    | LOSS   |
+-------+-----------------+---------+----------------+---------+-----------------+-------+-------+-------+--------+
To view other history, enter "Y":
```

Figure 32(5.2.2):Testcase7

Db output:



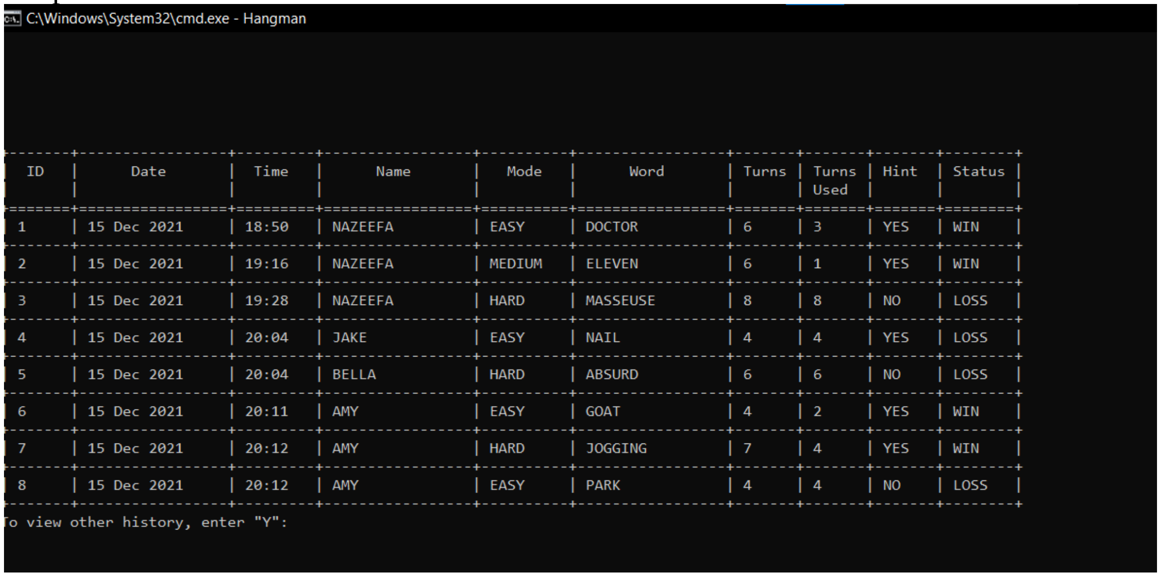
+ Options

	gameID	date	time	name	mode	word	turns	turnsUsed	hintGiven	winORloss
Edit Copy Delete	0001	15 Dec 2021	20:11	AMY	EASY	GOAT	4	2	YES	WIN
Edit Copy Delete	0002	15 Dec 2021	20:12	AMY	HARD	JOGGING	7	4	YES	WIN
Edit Copy Delete	0003	15 Dec 2021	20:12	AMY	EASY	PARK	4	4	NO	LOSS

Figure 33(5.2.3)Testcase8

2	2	(Table displaying 3 records)	(Table displaying 3 records)	Pass

Output:

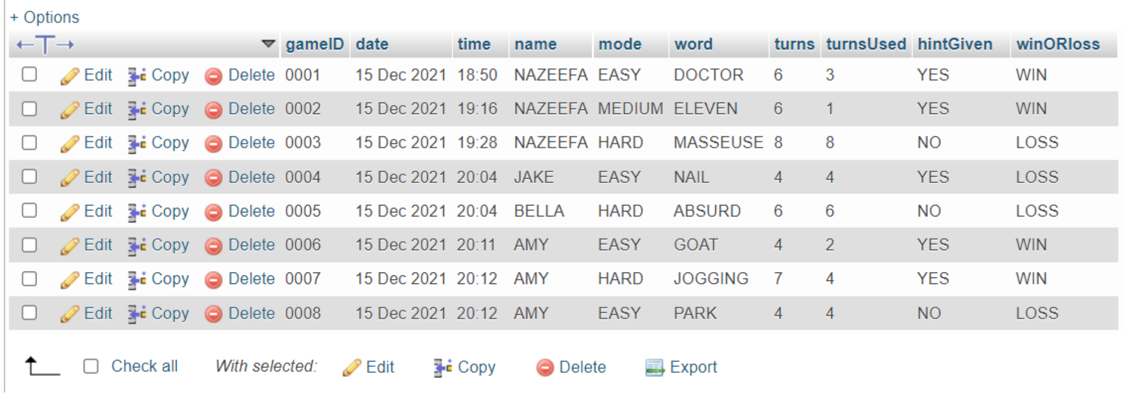


C:\Windows\System32\cmd.exe - Hangman

```
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 ID    |       Date        |   Time   |       Name        |   Mode    |       Word        | Turns  | Turns  | Hint   | Status  |
       |                   |          |                   |           |                   |        | Used   |        |         |
=======+===================+==========+===================+===========+===================+========+========+========+=========+
 1     | 15 Dec 2021       | 18:50    | NAZEEFA           | EASY      | DOCTOR            | 6      | 3      | YES    | WIN     |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 2     | 15 Dec 2021       | 19:16    | NAZEEFA           | MEDIUM    | ELEVEN            | 6      | 1      | YES    | WIN     |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 3     | 15 Dec 2021       | 19:28    | NAZEEFA           | HARD      | MASSEUSE          | 8      | 8      | NO     | LOSS    |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 4     | 15 Dec 2021       | 20:04    | JAKE              | EASY      | NAIL              | 4      | 4      | YES    | LOSS    |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 5     | 15 Dec 2021       | 20:04    | BELLA             | HARD      | ABSURD            | 6      | 6      | NO     | LOSS    |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 6     | 15 Dec 2021       | 20:11    | AMY               | EASY      | GOAT              | 4      | 2      | YES    | WIN     |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 7     | 15 Dec 2021       | 20:12    | AMY               | HARD      | JOGGING           | 7      | 4      | YES    | WIN     |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
 8     | 15 Dec 2021       | 20:12    | AMY               | EASY      | PARK              | 4      | 4      | NO     | LOSS    |
-------+-------------------+----------+-------------------+-----------+-------------------+--------+--------+--------+---------+
o view other history, enter "Y":
```

Figure 34(5.2.4):Testcase9

Db output:



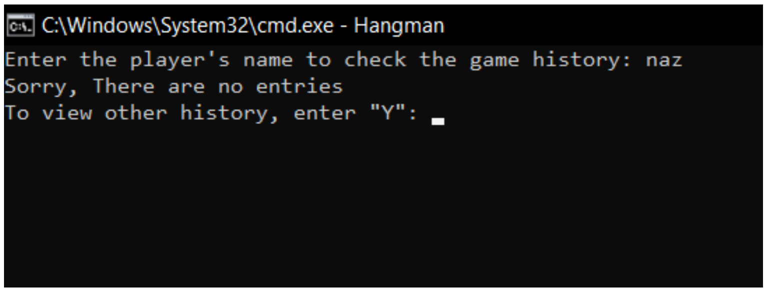
+ Options

	gameID	date	time	name	mode	word	turns	turnsUsed	hintGiven	winORloss
Edit Copy Delete	0001	15 Dec 2021	18:50	NAZEEFA	EASY	DOCTOR	6	3	YES	WIN
Edit Copy Delete	0002	15 Dec 2021	19:16	NAZEEFA	MEDIUM	ELEVEN	6	1	YES	WIN
Edit Copy Delete	0003	15 Dec 2021	19:28	NAZEEFA	HARD	MASSEUSE	8	8	NO	LOSS
Edit Copy Delete	0004	15 Dec 2021	20:04	JAKE	EASY	NAIL	4	4	YES	LOSS
Edit Copy Delete	0005	15 Dec 2021	20:04	BELLA	HARD	ABSURD	6	6	NO	LOSS
Edit Copy Delete	0006	15 Dec 2021	20:11	AMY	EASY	GOAT	4	2	YES	WIN
Edit Copy Delete	0007	15 Dec 2021	20:12	AMY	HARD	JOGGING	7	4	YES	WIN
Edit Copy Delete	0008	15 Dec 2021	20:12	AMY	EASY	PARK	4	4	NO	LOSS

Check all *With selected:* Edit Copy Delete Export

Figure 35(5.2.5):Testcase10

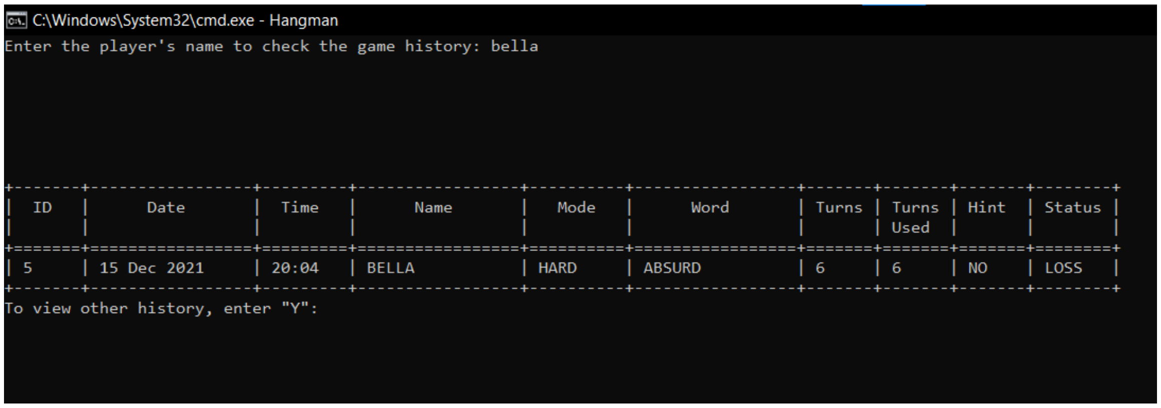
3	3 Name → naz	Sorry, There are no entries	Sorry, There are no entries	pass



```
C:\Windows\System32\cmd.exe - Hangman
Enter the player's name to check the game history: naz
Sorry, There are no entries
To view other history, enter "Y":
```

Figure 36(5.2.6):Testcase11

4	3 Name → Bella	(Table displaying 1 record)	(Table displaying 1 record)	pass

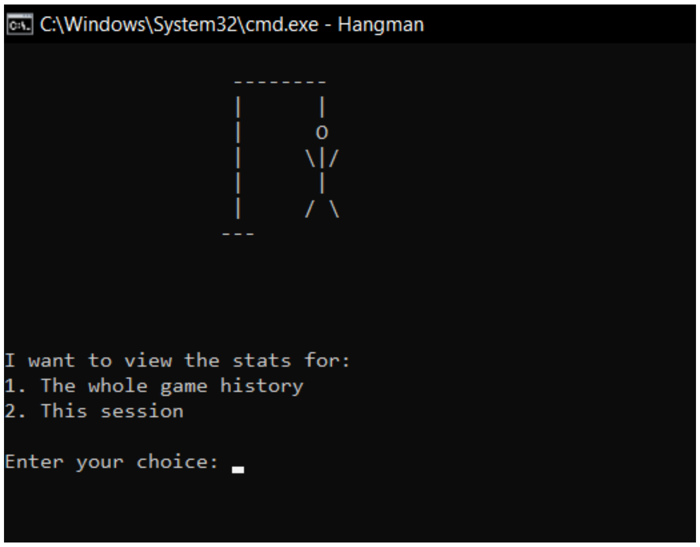
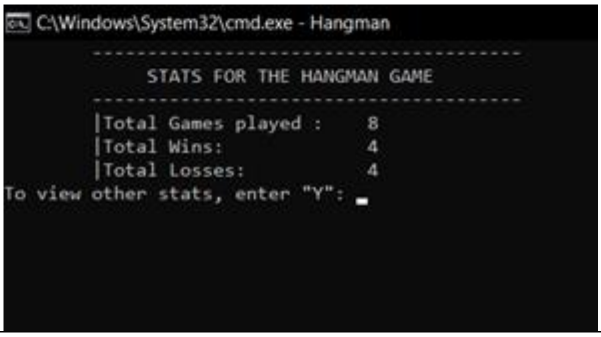


```
C:\Windows\System32\cmd.exe - Hangman
Enter the player's name to check the game history: bella




+-------+-----------------+---------+-----------------+----------+-----------------+-------+-------+-------+--------+
|  ID   |      Date       |  Time   |      Name       |   Mode   |      Word       | Turns | Turns | Hint  | Status |
|       |                 |         |                 |          |                 |       | Used  |       |        |
+=======+=================+=========+=================+==========+=================+=======+=======+=======+========+
| 5     | 15 Dec 2021     | 20:04   | BELLA           | HARD     | ABSURD          | 6     | 6     | NO    | LOSS   |
+-------+-----------------+---------+-----------------+----------+-----------------+-------+-------+-------+--------+
To view other history, enter "Y":
```

Figure 37(5.2.7):Testcase12

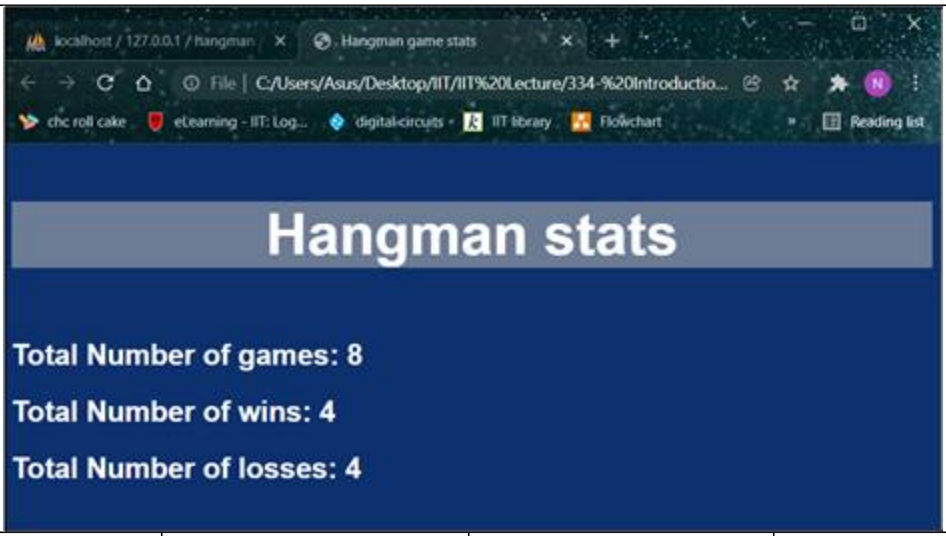
Table 2(5.2.1):Game history test case

5.3 Game stats

Test case #	Input	Expected output	Actual output	Remarks
	Choice for the Stats			
				
1	1	Total Games played : 8 Total Wins: 4 Total Losses: 4	Total Games played : 8 Total Wins: 4 Total Losses: 4	pass
	Output: 			
	Html output:			

```
C:\Windows\System32\cmd.exe - Hangman

                   --------
                   |      |
                   |      O
                   |     \|/
                   |      |
                   |     / \
                  ---



I want to view the stats for:
1. The whole game history
2. This session

Enter your choice: _
```

```
C:\Windows\System32\cmd.exe - Hangman
        ----------------------------------------
             STATS FOR THE HANGMAN GAME
        ----------------------------------------
        |Total Games played :    8
        |Total Wins:             4
        |Total Losses:           4
To view other stats, enter "Y": _
```



Hangman stats

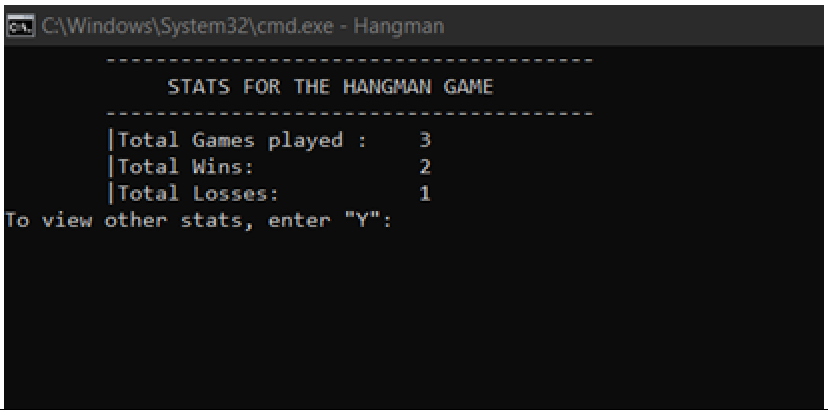
Total Number of games: 8

Total Number of wins: 4

Total Number of losses: 4

2	2	Total Games played : 3 Total Wins: 2 Total Losses: 1	Total Games played : 3 Total Wins: 2 Total Losses: 1	pass

Output:



```
        ---------------------------------------
             STATS FOR THE HANGMAN GAME
        ---------------------------------------
        |Total Games played :     3
        |Total Wins:              2
        |Total Losses:            1
To view other stats, enter "Y":
```

Html output:

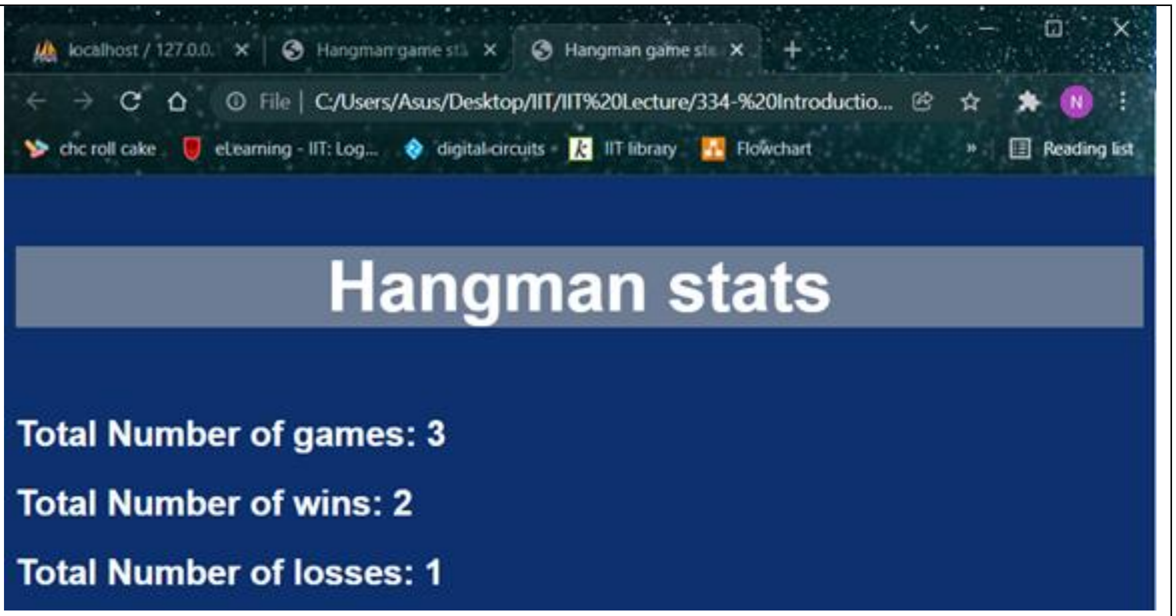
	 localhost / 127.0.0. Hangman game sta Hangman game sta File | C:/Users/Asus/Desktop/IIT/IIT%20Lecture/334-%20Introductio... chc roll cake eLearning - IIT: Log... digital-circuits IIT library Flowchart Reading list Hangman stats Total Number of games: 3 Total Number of wins: 2 Total Number of losses: 1

Table 3(5.3.1):stats test case

6. Conclusion

At the start of this report, a brief introduction to the task was given by stating the problem and the problem understanding. Then the code was explained with the usage of the packages, modules, and functions. Subsequently, the screenshots of the running program were attached and few test cases used for testing were also attached . In conclusion, this report discusses about the final program of the code that was created for the hangman game .

References

GeeksforGeeks. (n.d.). *Python Programming Language*. [online] Available at: https://www.geeksforgeeks.org/python-programming-language/?ref=shm [Accessed 10 Dec. 2021]

GeeksforGeeks. (2020). *TextTable module in Python*. [online] Available at: https://www.geeksforgeeks.org/texttable-module-in-python/ [Accessed 15 Dec. 2021].

www.a2hosting.com. (n.d.). *How to connect to MySQL using Python*. [online] Available at: https://www.a2hosting.com/kb/developer-corner/mysql/connecting-to-mysql-using-python [Accessed 15 Dec. 2021].

Individual statement

I confirm that I understand what plagiarism is. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged.

Signature: [illegible]

Date: 15.12.2021