SRI SATHYA SAI VIDYA VIHAR

AISSCE

2020-21

A Project Work in the subject

Computer Science for the

partial fulfillment of the

syllabus as prescribed by

CBSE for the session 2020-21

THE HANGMAN GAME

Submitted to:

Ms. Jaspal Vishwakarma

Submitted by:

Sanskar Sharma

CONTENTS

CERITIFICATE

This is to certify that **Sanskar Sharma** of class **XII-B** has completed his Project work as per the syllabus and has submitted a satisfactory account of it as a part of fulfillment towards the practical course for All India Senior Secondary Certificate Examination, 2020-21.

Roll no. Date:_____

_____ _____

(Internal Examiner) (External Examiner)

Ms. Jaspal Vishwakarma

_____

(**Principal**)

Ms. Punita Nehru **School Seal**

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Ms. Jaspal Vishwakarma as well as our Principal ma'am Ms. Punita Nehru who gave me the golden opportunity to do this wonderful project, which also helped me in doing a lot of research and I came to know about so many new and fascinating things. Therefore I am really thankful to them. Secondly i would also like to thank my parents and friends who helped me in finalizing this project within the limited time frame.

CERTIFICATE OF ORIGINALITY

This is to certify that is an original work of the student and is being submitted as a

part of fulfillment towards the practical course of All India Senior Secondary Certificate Examination 2020-21.

This report has not been submitted earlier either to the Institute or to any other institution for the fulfillment of the requirement of a course of study.

Jaspal Vishwakarma **Sanskar Sharma**

Project Guide **Student**

AIM OF THE PROJECT

Aim of the project is to provide an electronic version that could be used to play the beautiful game without using papers, which would help us to save our resources such as trees used to produce paper and our precious time which we spend to write the names of different topics and words. The game will definitely enhance the thinking skills of players as the game is all about hints and guesses. Overall it’s a complete fun zone which is not addictive and is related to English and words that helps us to upgrade our responses.

INTRODUCTION TO PROJECT

The project is to design the famous game Hangman on computer using Python and SQL. In this project we ask the player to guess a four letter (level-1) or a five letter (level-2) word depending on the level, if he answers incorrectly then after two chances a hint is displayed that gives the name of topic of word and if he answers incorrectly his second last chance then again a hint is displayed that tells the player something special about the given word. After each wrong guess parts of hangman are formed simultaneously and a complete image of it is displayed if the player loses the game. The words and their two hints are stored in SQL which are linked with Python and randomly executed .The game has option to enable user to continue his game even after closing the playing window as the details can be stored in python according to the player’s wish.

INTRODUCTION TO MODULES

- TKINTER MODULE: **Tkinter** is a Python binding to the Tk GUI toolkit. Tk is the original GUI library for the Tcl language. **Tkinter** is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. It helps us to present our code in a user friendly way.
- RANDOM: This module provides functions for generating pseudo-random numbers or variable.
- MYSQL.CONNECTOR: **MySQL Connector/Python** allows you to compress the data stream between **Python** and **MySQL** database server using protocol compression. It supports connections using TCP/IP socket and secure TCP/IP connection using SSL. **MySQLConnector/Python** is an API implemented using pure **Python**.
- COLLECTIONS: **Collections** in Python are containers that are used to store **collections** of data, for example, list, dict, set, tuple etc. These are built-in **collections**. Several **modules** have been developed that provide additional data structures to store **collections** of data.
- DATETIME: The **datetime module** supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.
- PYGAME: **Pygame** is a cross-platform set of **Python modules** designed for writing video games. It includes computer graphics and sound libraries designed to be used with the **Python** programming language.

DATA DICTIONARY

----------CLASS playgame ----------

VARIABLE	DESCRIPTION
globnum	This number is level of the game.
given_id	The id of the player if new is created.
playerid	The id of the player if user resume next round.
score	Score of the game.

MEMBER FUNCTIONS	DESCRIPTION
main_working	It works for the guessed letter and button function.
work	It is main function for the working of the game.
bestscore_function	It stores the best score of the user in the MYSQL when the best score broken.
button_function	It helps the entry button.
button_playagain	It works to set the level.
play	This gets the words and hints from the database.
level	It helps set the level of the game accordingly.

----------CLASS player_entry ----------

VARIABLE	DESCRIPTION
New	This tells us if new player is created(True/False variable).
Resume	This tells us if the player has resumed(True/False variable).
Save	This tells us if the player has continued a saved game(True/False variable).
MEMBER FUNCTIONS	DESCRIPTION
create_player	To enter the created player info into MYSQL.
resume_player	To get the resumed player info to check player is there or not.
saved_game	To check if there is a saved game.

delete_savedgame	Deleted the saved game when the user starts the saved game.
save_play	To continue the saved game if any.

----------CLASS player_sql_entry ----------

VARIABLE	DESCRIPTION
playerid_checkvariable	Given player id to check it exists in MYSQL records.
playername_checkvariable	Given player name to check it exists in MYSQL records.

MEMBER FUNCTION	DESCRIPTION
create_new_player(s)	It is the window function where new players are created(details are given here).
resume_old_player(s)	It is the window function where played can play the next game(details are given here).
saved_play(s)	It is the window function where user can play the saved game, (details are given here).

----------Database----------

player_table

Field Name	Data type	Brief
Player ID	Int	Primary key –used to link with score table
Player Name	Varchar	Player name user input
Player Age	Int	Player age
Player Gender	Varchar	Player gender

hint_table

Field Name	Data type	Brief
Topic	Varchar	Topic of the word.
Word	Varchar	Word
Hint1	Varchar	Hint1 of the word.
Hint2	Varchar	Hint2 of the word.
Levelnum	Int	Level number.

record_table

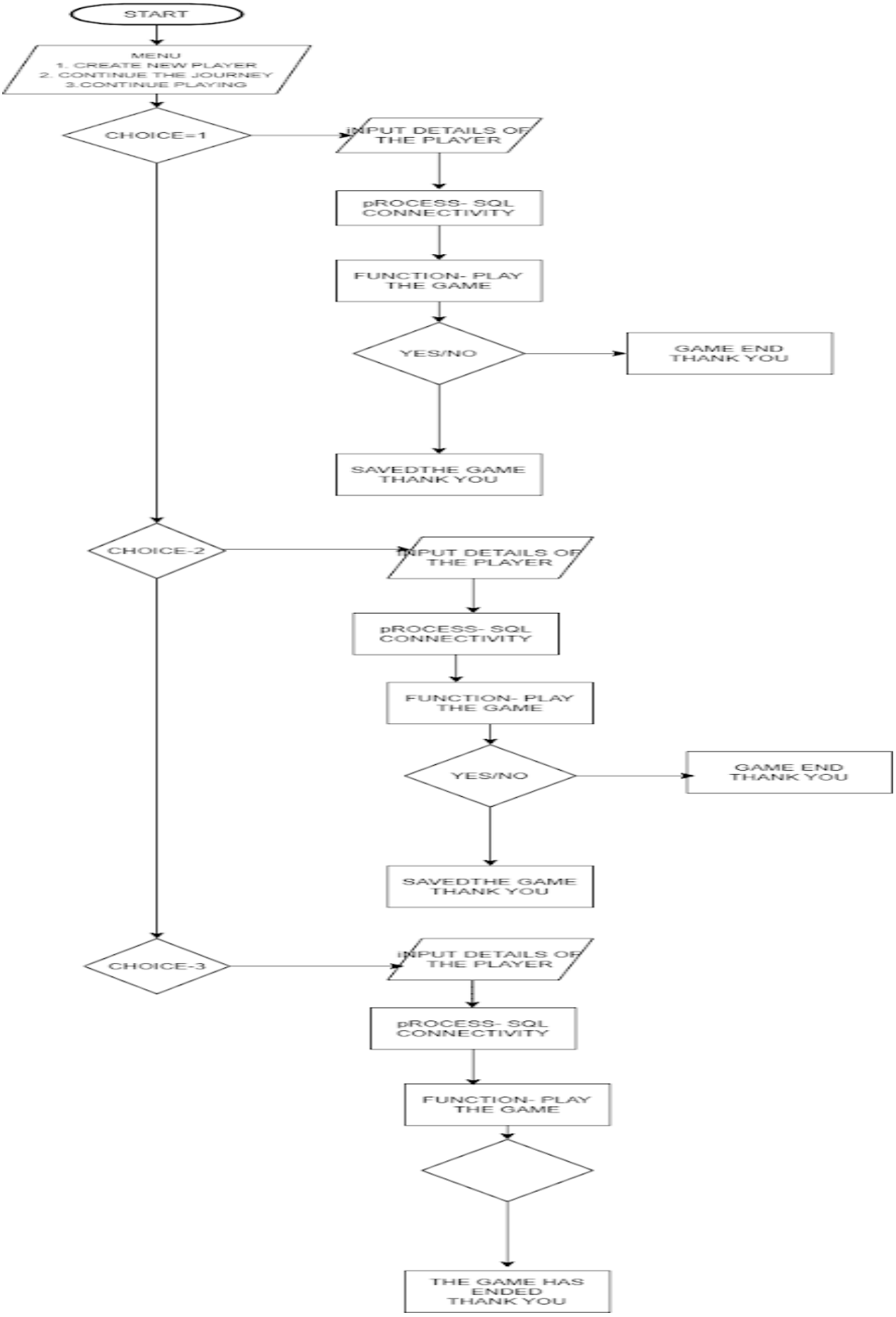
Field Name	Data type	Brief
Player ID	Int	Foreign key –helps to link with the master table.
Best score	Int	It only updates if the past score for that player is beaten.
Date of the best score	Date time	Shows the time of their achievement .

saved_game

Field Name	Data type	Brief
Player ID	Int	Helps to connect to both the above table.
Playername	Int	Player name.
Saved_word	Int	Word on which user was
Levelplayed	Int	

score_game

Field Name	Data type	Brief
Player ID	Int	Helps to connect to both the above table.
Number of level passed	Int	Shows how many levels passed.
Total score	Int	Total round score.


START
MENU
1. CREATE NEW PLAYER
2. CONTINUE THE JOURNEY
3.CONTINUE PLAYING
CHOICE=1
INPUT DETAILS OF THE PLAYER
pROCESS- SQL CONNECTIVITY
FUNCTION- PLAY THE GAME
YES/NO
GAME END THANK YOU
SAVEDTHE GAME THANK YOU
CHOICE-2
INPUT DETAILS OF THE PLAYER
pROCESS- SQL CONNECTIVITY
FUNCTION- PLAY THE GAME
YES/NO
GAME END THANK YOU
SAVEDTHE GAME THANK YOU
CHOICE-3
INPUT DETAILS OF THE PLAYER
pROCESS- SQL CONNECTIVITY
FUNCTION- PLAY THE GAME
THE GAME HAS ENDED THANK YOU

SOURCE CODE

```
"""Please save the changes that you have done to Google drive also.thank you ."""

#--------Legal_Import_statement_section-------

import tkinter as tk

from tkinter import *

from tkinter import PhotoImage

from tkinter import messagebox

from tkinter import ttk

import random

from collections import Counter

import mysql.connector

import datetime

import pygame

 #***************SQL_DETAILS***************

  """|^_^|change the details according to your system|^_^|!"""

 global sql_host,sql_pass,sql_database,sql_user

sql_host="localhost"

sql_pass="manmandir"

sql_database="hangman(player details)"

sql_user="root"

 #-------------Create_main_window----------------

window=tk.Tk()

window.title("Hangman|welcome|")

icon=tk.PhotoImage(file ="hangmanicon3.png")

window.iconphoto(False,icon)

window.geometry("800x550")

window.configure(bg="navy")

pygame.mixer.init()

 #-----------------Main_program----------------------

class playgame:

        def __init__(s):

                s.score=0

                s.correct = 0

        def main_working(s):

                s.letterGuessed = ''# list for storing the letters guessed by the player

                s.chances =8

                s.flag = 0

                s.wrong_count=0

                playgame. button_function(s)

        def work(s):

                s.flag = 0

                s.hint_used=0

                s.guess_string=s.guess.get()

                if (s.chances != 0) and s.flag == 0: #flag is updated when the word is correctly guessed

                        # Validation of the guess

                        if s.guess_string.isalpha()==False:

                                messagebox.showwarning("Message",'Enter only a LETTER')

                        elif len(s.guess_string) > 1:

                                messagebox.showwarning("Message",'Enter only a SINGLE letter')
```

```
                elif s.guess_string in s.guessedletter:

messagebox.showinfo("Message",'You have already guessed that letter')

                global label_letterguessed

s.guessedletter=s.guessedletter+s.guess_string+","

label_letterguessed.config(text="Letters Used:"+s.guessedletter)

                # If letter is guessed correctly

                if s.guess_string in s.word:

                        s.k = s.word.count(s.guess_string) #k stores the number of times the guessed letter occurs in the word

                        for _ in range(s.k):

                                s.letterGuessed += s.guess_string# The guess letter is added as many times as it occurs

                # Print the word

                s.x=Counter(s.letterGuessed)

                s.y=Counter(s.word)

                wrong=False#to check if wrong guess

                        newword=""

                        for char in s.word:

                                if char in s.letterGuessed and (s.x != s.y):#going rt on track

                                        if char==s.guess_string:

                                                global label__

                                                newword=""

                                                for i in s.word:#required for the word to update on the screen

                                                        if char ==i:

                                                                newword=newword+char# For printing the empty spaces for letters of the word

                                                        elif i in s.letterGuessed and char!=i :

                                                                newword=newword+i

                                                        else:

                                                                newword=newword+'_ '

                                                label__.config(text=newword)

                                                s.correct += 1

                # If user has guessed all the letters

                elif  s.x == s.y  : # Correct Guessed

                        label__.config(text=s.word)

                        global globnum

                        globnum+=1

s.buton_entry.config(state=DISABLED)

label_word=tk.Label(s.mainwindow,text="Congratulation, you saved the hangman|^o^|"+"The word was: "+s.word,fg="blue",bg="orange",font=("Ink Free",16,"bold")).pack()

                s.flag = 1

                global sql_host,sql_pass,sql_database,sql_user

db_conn=mysql.connector.connect(

                host=sql_host,

                user=sql_user,
```

```
                passwd=sql_pass,
                database= sql_database)
                s.score=s.correct
            -s.hint_used
                global playerid,given_id
                global resume,new
                c=db_conn.cursor()
                q="INSERT INTO
SCORE_TABLE VALUES(%s,%s,%s)"
                if resume== True:
    data=(given_id,globnum,s.score)
                elif new==True:
    data=(playerid,globnum,s.score)
                c.execute(q,data)
                db_conn.commit()
                c.close()
                db_conn.close()

player_sql_entry.delete_savedgame(s)

    playgame.bestscore_function(s)

    playgame.button_playagain(s)
                break# To break out of the
                for loop
                return
                else :
                wrong=True
            if s.guess_string not in s.word
          and wrong== True:
                s.chances = s.chances -1
                s.wrong_count+=1
                if s.wrong_count==1:

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)
                elif s.wrong_count==2 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)
                global hint1,label_hint

label_hint=tk.Label(s.mainwindow,text=hint1,
fg="blue",bg="orange",font=("Ink
Free",15,"bold"))
                label_hint.pack()
                s.hint_used=1
                elif s.wrong_count==3 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)
                elif s.wrong_count==4 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)
```

```
s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

elif s.wrong_count==5 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)

s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

s.canvas_main.create_oval(175,175,225,225,w
idth=3,fill="orange")#head

elif s.wrong_count==6 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)

s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

s.canvas_main.create_oval(175,175,225,225,w
idth=3,fill="orange")#head

s.canvas_main.create_line(200,225,200,275,fil
l="orange",width=3)#body

s.canvas_main.create_line(200,225,150,175,fil
l="orange",width=3)#hand1

global hint2

label_hint.config(text=hint2)

s.hint_used=2

elif s.wrong_count==7 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)

s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

s.canvas_main.create_oval(175,175,225,225,w
idth=3,fill="orange")#head

s.canvas_main.create_line(200,225,200,275,fil
l="orange",width=3)#body

s.canvas_main.create_line(200,225,150,175,fil
l="orange",width=3)#hand

s.canvas_main.create_line(200,225,250,175,fil
l="orange",width=3)

elif s.wrong_count==8 :

s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)
```

```
s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

s.canvas_main.create_oval(175,175,225,225,w
idth=3,fill="orange")

s.canvas_main.create_line(200,225,200,275,fil
l="orange",width=3)

s.canvas_main.create_line(200,225,150,175,fil
l="orange",width=3)#hand

s.canvas_main.create_line(200,225,250,175,fil
l="orange",width=3)#hand

s.canvas_main.create_line(200,275,250,290,fil
l="orange",width=3)#leg

s.canvas_main.create_line(200,275,150,290,fil
l="orange",width=3)#leg

                if s.chances <= 0 and (s.x != s.y):

                        globnum+=1

label_lost=tk.Label(s.mainwindow,text="You
Lost.Better Luck Next Time!"+s.word+" was
the
word.",fg="blue",bg="orange",font=("Freestyle
Script",16,"bold")).pack()


s.buton_entry.config(state=DISABLED)

                playgame.button_playagain(s)


s.canvas_main.create_line(100,300,300,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,100,300,fil
l="orange",width=3)

s.canvas_main.create_line(100,100,200,100,fil
l="orange",width=3)
```

```
s.canvas_main.create_line(200,100,200,175,fil
l="orange",width=3)

s.canvas_main.create_oval(175,175,225,225,fi
ll="orange",width=2)#head

s.canvas_main.create_line(200,225,200,275,fil
l="orange",width=3)#body

s.canvas_main.create_line(200,225,150,175,fil
l="orange",width=3)#hand

s.canvas_main.create_line(200,225,250,175,fil
l="orange",width=3)#hand

s.canvas_main.create_line(200,275,250,290,fil
l="orange",width=3)#leg

s.canvas_main.create_line(200,275,150,290,fil
l="orange",width=3)#leg


    player_sql_entry.delete_savedgame(s)

                return

    def bestscore_function(s):#the best score
function to store the former.

                global
sql_host,sql_pass,sql_database,sql_user

        db_conn=mysql.connector.connect(

                        host=sql_host,

                        user=sql_user,

                    passwd=sql_pass,

                database=sql_database)

        global resume,new,playerid,given_id

        c=db_conn.cursor(buffered=True)

        best_datetime=datetime.datetime.now()
```

```
best_date=datetime.date(best_datetime.year,best_datetime.month,best_datetime.day)

if resume==True:

data=(given_id,)

q="SELECT BESTSCORE FROM RECORD_TABLE WHERE PLAYERID=(%s)"

c.execute(q,data)

bestscore=c.fetchone( )

if s.score>bestscore[0]:

q="UPDATE RECORD_TABLE  SET BESTSCORE=%s,SCOREDATE=%s WHERE PLAYERID=%s"


data=(s.score,best_date,given_id)

elif new==True:

data=(playerid,)

q="SELECT BESTSCORE FROM RECORD_TABLE WHERE PLAYERID=(%s)"

c.execute(q,data)

bestscore=c.fetchone( )

if s.score>bestscore[0]:

q="UPDATE RECORD_TABLE  SET BESTSCORE=%s,SCOREDATE=%s WHERE PLAYERID=%s"


data=(s.score,best_date,playerid)

c.execute(q,data)

db_conn.commit()

c.close()

db_conn.close()

def button_function(s):

s.guess_variable=tk.StringVar()

s.guess= ttk.Entry(s.mainwindow,textvariable=s.guess_variable,width=10,justify="center")

s.guess.pack()


s.buton_entry=tk.Button(s.mainwindow,text="Enter",command= lambda:playgame.work(s),fg="navy",bg="orange",font=("small fonts",16,"bold"),activeforeground="orange",activebackground="navy")

s.buton_entry.pack()

def button_playagain(s):

if globnum==1:


buton_again=tk.Button(s.mainwindow,text="Next Round!",command= lambda:playgame.level(s,1),fg="blue",bg="orange",font=("Freestyle Script",16,"bold"),activeforeground="blue",activebackground="blue").pack()

elif globnum==2 or globnum==3:


buton_again=tk.Button(s.mainwindow,text="Next Round!",command= lambda:playgame.level(s,2),fg="blue",bg="orange",font=("Freestyle Script",16,"bold"),activeforeground="blue",activebackground="blue").pack()

else:

s.mainwindow.destroy()

def play(s,num_of_letters):

global mainwindow

s.mainwindow=tk.Tk()

s.mainwindow.title("Hangman|Play area|")

s.mainwindow.geometry("700x700")

s.mainwindow.configure(bg="orange")
```

```
label_hangman=tk.Label(s.mainwindow,text="HANGMAN",font=("small fonts",50),fg="navy",bg="orange").pack()

global sql_host,sql_pass,sql_database,sql_user

db_conn=mysql.connector.connect(

host=sql_host,

user=sql_user,

passwd=sql_pass,

database=sql_database)

c=db_conn.cursor()

q="SELECT * FROM HINT_TABLE WHERE LEVELNUM=(%s)"

data=(num_of_letters,)

c.execute(q,data)

rows=c.fetchmany(60)

n1=random.randint(1,len(rows))#can leave the empty it reads one record (doubt!)

global saved

global globalword

if saved==True:

s.word=globalword#varialbe to have the saved word

else:

try:

s.word=rows[n1][1]

except:


messagebox.showwarning("Message",'|#_@|Sorry our hangman ran off . Please try again.|@_#|')

#The main canvas for mr.hangman

s.canvas_main=tk.Canvas(s.mainwindow,width=350,height=350,background="navy")


s.canvas_main.pack()#s.canvas_main.create_line(x1,y1,x2,y2,fill="colour")

global hint1,hint2

hint1=rows[n1][2]#basic hint to tell about the word type

hint2=rows[n1][3]#a more related hint

global label__,label_letterguessed

blankword=""

s.guessedletter=""

for i in s.word:

blankword=blankword+'_ ' # For printing the empty spaces for letters of the word


label__=tk.Label(s.mainwindow,text=blankword,fg="blue",bg="orange",font=("Freestyle Script",25))

label__.pack()


label_letterguessed=tk.Label(s.mainwindow,text="Guessed Letters: ",fg="blue",bg="orange",font=("Freestyle Script",20))

label_letterguessed.pack()

button_save= Button(s.mainwindow, text = "Save!",command=lambda:player_sql_entry.save_play(s,num_of_letters,s.word),fg="navy",bg="orange",font=("small fonts",16,"bold"),activeforeground="orange",activebackground="navy")

button_save.place(relx = 1, x =-2, y = 2, anchor = NE)

playgame.main_working(s)

s.mainwindow.mainloop()
```

```
def level(s,level_num):
if level_num==1:
playgame.play(s,1)
elif level_num==2:
playgame.play(s,2)
#----------Connecting_to_MYSQL----------
"""|^_^|main sql conections ahead|^_^|!"""
class player_sql_entry :
def __init__(s):
s.rid=100
def create_player(s):
global playername_variable, playerage_variable,playergender_variable,button_submit,new
new=True
button_submit.config(state=DISABLED)
name=playername_variable.get()
age=playerage_variable.get()
gender=playergender_variable.get()
p=playgame()
global sql_host,sql_pass,sql_database,sql_user
db_conn=mysql.connector.connect(
host=sql_host,
user=sql_user,
passwd=sql_pass,
database= sql_database)
s.num= random.randint(10,900)
global playerid
playerid=100+s.num+int(age)
c=db_conn.cursor()
q="INSERT INTO  PLAYER_TABLE VALUES(%s,%s,%s,%s)"
data=(playerid,name,age,gender)
c.execute(q,data)
db_conn.commit()
c.close()
db_conn.close()
db_conn=mysql.connector.connect(
host=sql_host,
user=sql_user,
passwd=sql_pass,
database= sql_database)
c=db_conn.cursor()
best_datetime=datetime.datetime.now()
best_date=datetime.date(best_datetime.year,best_datetime.month,best_datetime.day)
q="INSERT INTO RECORD_TABLE VALUES(%s,%s,%s)"
data=(playerid,0,best_date)
c.execute(q,data)
db_conn.commit()
c.close()
db_conn.close()
s.create_new_window2=tk.Tk()
s.create_new_window2.title("Hangman|Create Player|")
s.create_new_window2.geometry("300x300")
s.create_new_window2.configure(bg="orange")
```

```
label_id=tk.Label(s.create_new_window2,text
="YourID:"+str(playerid),font=("small
fonts",17),fg="navy",bg="orange").pack()

label_rem=tk.Label(s.create_new_window2,te
xt="Please remember your ID!",font=("small
fonts",12),fg="navy",bg="orange").pack()

button_play=Button(s.create_new_window2,t
ext="Play",command=lambda:
p.level(1),bg="navy",fg="orange")

button_play.pack()

s.create_new_window2.mainloop()

def resume_player(s):

p=playgame()

global
playerid_checkvariable,playername_checkvari
able

global given_id,resume

resume=True

given_id=playerid_checkvariable.get()

given_name=playername_checkvariable.get()

global
sql_host,sql_pass,sql_database,sql_user

db_conn=mysql.connector.connect(

host=sql_host,

user=sql_user,

passwd=sql_pass,

database= sql_database)

c1=db_conn.cursor()

q1="SELECT PLAYERNAME FROM
PLAYER_TABLE  WHERE PLAYERID = %s ;"

data=(given_id,)

c1.execute(q1,data)

row=c1.fetchone()

db_conn.commit()

c1.close()

db_conn.close()

if row!=None:

if row[0]!= None  and  row[0] ==
given_name:

global button_continue

button_continue.config(state=DISABLED)

s.create_new_window2=tk.Tk()

s.create_new_window2.title("Hangman|Saved
Player|")

s.create_new_window2.geometry("200x150")

s.create_new_window2.configure(bg="orange
")

label_hangman=tk.Label(s.create_new_windo
w2,text="HANGMAN",font=("small
fonts",29),fg="navy",bg="orange").pack()

label_welcome=tk.Label(s.create_new_windo
w2,text="Welcome "+given_name+"
!",font=("small
fonts",17),fg="navy",bg="orange").pack()

button_play=Button(s.create_new_window2,t
ext="Play",command=lambda:
p.level(1),fg="navy",bg="orange").pack()

s.create_new_window2.mainloop()

else:

s.create_new_window2=tk.Tk()

s.create_new_window2.title("Hangman|Creat
e Player|")
```

```
s.create_new_window2.geometry("400x150")

s.create_new_window2.configure(bg="orange
")

label_hangman=tk.Label(s.create_new_windo
w2,text="HANGMAN",font=("small
fonts",50),fg="navy",bg="orange").pack()

label_welcome=tk.Label(s.create_new_windo
w2,text="Sorry wrong username
!",font=("small
fonts",17),fg="navy",bg="orange").pack()

        s.create_new_window2.mainloop()

        else:

messagebox.showwarning("Message",'|#_@|Y
ou have enter a wrong ID or username|@_#|')

    def saved_game(s):

        p=playgame()

        global
playerid_checkvariable,playername_checkvari
able,given_id,resume

        resume=True

        given_id=playerid_checkvariable.get()

given_name=playername_checkvariable.get()

        global
sql_host,sql_pass,sql_database,sql_user

        db_conn=mysql.connector.connect(

            host=sql_host,

            user=sql_user,

            passwd=sql_pass,

            database= sql_database)

        c1=db_conn.cursor()

        q1="SELECT
PLAYERNAME,SAVED_WORD,LEVELPLAYED
FROM  SAVED_GAME  WHERE PLAYERID =
%s ;"

        data=(given_id,)

        c1.execute(q1,data)

        row=c1.fetchone()

        db_conn.commit()

        c1.close()

        db_conn.close()

        if row!=None:

            global globalword,saved

            globalword=row[1]

            saved=True

        if row!= None  and  row[0] ==
given_name:

            global button_continue


button_continue.config(state=DISABLED)

        s.create_new_window2=tk.Tk()

s.create_new_window2.title("Hangman|Creat
e Player|")

s.create_new_window2.geometry("200x150")

s.create_new_window2.configure(bg="orange
")

label_hangman=tk.Label(s.create_new_windo
w2,text="HANGMAN",font=("small
fonts",29),fg="navy",bg="orange").pack()

label_welcome=tk.Label(s.create_new_windo
w2,text="Welcome "+given_name+"
!",font=("small
fonts",17),fg="navy",bg="orange").pack()
```

```
button_play=Button(s.create_new_window2,text="Play",command=lambda: p.level(1),fg="navy",bg="orange").pack()
                s.create_new_window2.mainloop()
            else:
                s.create_new_window2=tk.Tk()

s.create_new_window2.title("Hangman|Create Player|")

s.create_new_window2.geometry("400x150")

s.create_new_window2.configure(bg="orange")

label_hangman=tk.Label(s.create_new_window2,text="HANGMAN",font=("small fonts",50),fg="navy",bg="orange").pack()

label_welcome=tk.Label(s.create_new_window2,text="Sorry wrong username !",font=("small fonts",17),fg="navy",bg="orange").pack()
                s.create_new_window2.mainloop()
            else:

messagebox.showwarning("Message",'|#_@|You have enter a wrong ID or username|OR|You donot have a saved game|@_#|')
        def delete_savedgame(s):
            global saved
            global given_id,resume,new,playerid
            if saved==True:

db_conn=mysql.connector.connect(
                    host=sql_host,
                    user=sql_user,
                    passwd=sql_pass,
                    database= sql_database)
                c=db_conn.cursor()
                q="DELETE SAVED_GAME WHERE PALYERID=(%s)"
                if resume== True:
                    data=(given_id,)
                elif new==True:
                    data=(playerid,)
                c.execute(q,data)
                db_conn.commit()
                c.close()
                db_conn.close()
                saved=False
        def save_play(s,level,word):
            global sql_host,sql_pass,sql_database,sql_user
            db_conn=mysql.connector.connect(
                host=sql_host,
                user=sql_user,
                passwd=sql_pass,
                database= sql_database)
            c1=db_conn.cursor()
            q1="INSERT INTO SAVED_GAME VALUES(%s,%s,%s,%s);"
            global playerid_checkvariable,playername_checkvariable
            global resume,new
            global globnum,playerid,given_id
```

if resume==True:

given_id=playerid_checkvariable.get()

given_name=playername_checkvariable.get()

data=(given_id,given_name,word,level)

elif new==True:

global playerid,playername_variable

given_id=playerid

data=(playerid,given_name,word,level)

c1.execute(q1,data)

db_conn.commit()

c1.close()

db_conn.close()

global window,mainwindow,create_new_window2

window.destroy()

s.mainwindow.destroy()

exit()

#Creating_New_Player|OR|Resume_Old_Game

"""|^_^|Entry code of game.kind of like a login screen for the user|@_@|!"""

class player_entry :

def __init__(s):

s.something=0

def create_new_player(s):

global b_createplayer,b_resume,b_savedplay

b_createplayer.config(state=DISABLED)

b_resume.config(state=DISABLED)

b_savedplay.config(state=DISABLED)

global new

new=True

s.create_new_window=tk.Tk()

s.create_new_window.title("Hangman|Create Player|")

s.create_new_window.geometry("400x150")

s.create_new_window.configure(bg="navy")

global playername_variable, playerage_variable,playergender_variable

playername_variable=tk.StringVar(s.create_new_window)

playerage_variable=tk.IntVar(s.create_new_window)

playergender_variable=tk.StringVar(s.create_new_window)

s.label_name=tk.Label(s.create_new_window,text="Name:",font=("small fonts",17),fg="orange",bg="navy").grid(column=1,row=1)

s.player_name= tk.Entry(s.create_new_window,textvariable=playername_variable,width=17,justify="center",font=("small fonts",17),bg="orange",fg="navy")

s.player_name.grid(column=2,row=1)

s.label_age=tk.Label(s.create_new_window,text="Age:",font=("small fonts",17),fg="orange",bg="navy").grid(column=1,row=2)

s.player_age= tk.Entry(s.create_new_window,textvariable=playerage_variable,width=17,justify="center",f

```
ont=("small
fonts",17),bg="orange",fg="navy")

s.player_age.grid(column=2,row=2)


s.label_gender=tk.Label(s.create_new_windo
w,text="Gender:",font=("small
fonts",17),fg="orange",bg="navy").grid(colum
n=1,row=3)

s.player_gender=
tk.Entry(s.create_new_window,textvariable=p
layergender_variable,width=17,justify="cente
r",font=("small
fonts",17),bg="orange",fg="navy")

s.player_gender.grid(column=2,row=3)

global button_submit


button_submit=Button(s.create_new_window,
text="Submit",command=lambda:
player_sql_entry.create_player(s),bg="navy",fg
="orange")

button_submit.grid(column=2,row=4)

s.create_new_window.mainloop()

def resume_old_player(s):

global
b_createplayer,b_resume,b_savedplay

b_createplayer.config(state=DISABLED)

b_resume.config(state=DISABLED)

b_savedplay.config(state=DISABLED)

global resume

resume=True

p=playgame()

s.create_new_window3=tk.Tk()


s.create_new_window3.title("Hangman|Resu
me Player|")


s.create_new_window3.geometry("300x150")

s.create_new_window3.configure(bg="orange
")

global
playerid_checkvariable,playername_checkvari
able


playerid_checkvariable=tk.IntVar(s.create_ne
w_window3)


playername_checkvariable=tk.StringVar(s.cre
ate_new_window3)


s.label_age=tk.Label(s.create_new_window3,t
ext="Your ID:",font=("small
fonts",17),fg="navy",bg="orange").grid(colum
n=1,row=2)

s.playerid_check=
tk.Entry(s.create_new_window3,textvariable=
playerid_checkvariable,font=("small
fonts",13),bg="navy",fg="orange",width=17,ju
stify="center")

s.playerid_check.grid(column=2,row=2)


s.label_gender=tk.Label(s.create_new_windo
w3,text="Your name:",font=("small
fonts",17),fg="navy",bg="orange").grid(colum
n=1,row=3)

s.playername_check=
tk.Entry(s.create_new_window3,textvariable=
playername_checkvariable,font=("small
fonts",15),bg="navy",fg="orange",width=17,ju
stify="center")


s.playername_check.grid(column=2,row=3)

global button_continue


button_continue=Button(s.create_new_windo
w3,text="Continue",command=lambda:
player_sql_entry.resume_player(s),font=("sma
ll
fonts",15),bg="orange",fg="navy",activebackg
round="navy",activeforeground="orange")
```

```
button_continue.grid(column=2,row=4)

s.create_new_window3.mainloop()

def saved_play(s):

global
b_createplayer,b_resume,b_savedplay

b_createplayer.config(state=DISABLED)

b_resume.config(state=DISABLED)

b_savedplay.config(state=DISABLED)

global resume

resume=True

p=playgame()

s.create_new_window3=tk.Tk()


s.create_new_window3.title("Hangman|Conti
nue Saved|")


s.create_new_window3.geometry("300x150")


s.create_new_window3.configure(bg="orange
")

global
playerid_checkvariable,playername_checkvari
able


playerid_checkvariable=tk.IntVar(s.create_ne
w_window3)


playername_checkvariable=tk.StringVar(s.cre
ate_new_window3)


s.label_age=tk.Label(s.create_new_window3,t
ext="Your ID:",font=("small
fonts",17),fg="navy",bg="orange").grid(colum
n=1,row=2)

s.playerid_check=
tk.Entry(s.create_new_window3,textvariable=
playerid_checkvariable,font=("small
fonts",13),bg="navy",fg="orange",width=17,ju
stify="center")

s.playerid_check.grid(column=2,row=2)


s.label_gender=tk.Label(s.create_new_windo
w3,text="Your name:",font=("small
fonts",17),fg="navy",bg="orange").grid(colum
n=1,row=3)

s.playername_check=
tk.Entry(s.create_new_window3,textvariable=
playername_checkvariable,font=("small
fonts",15),bg="navy",fg="orange",width=17,ju
stify="center")


s.playername_check.grid(column=2,row=3)

global button_continue


button_continue=Button(s.create_new_windo
w3,text="Continue",command=lambda:
player_sql_entry.saved_game(s),font=("small
fonts",15),bg="orange",fg="navy",activebackg
round="navy",activeforeground="orange")

button_continue.grid(column=2,row=4)

s.create_new_window3.mainloop()

#-----------------Front_Page_code----------------

logo_1=tk.PhotoImage(file="hangman1.png")

logo_label=ttk.Label(window
,image=logo_1,border=0).pack()

p=player_entry()

global b_createplayer,b_resume,b_savedplay

b_createplayer=Button(window,text="Create
New Player",command=lambda:
p.create_new_player(),bg="navy",fg="orange",
font=("small
fonts",20),activeforeground="navy",activebac
kground="orange",border=0)

b_createplayer.pack()

b_resume=Button(window,text="Countinue
The Journey",command=lambda:
```

p.resume_old_player(),font=("small fonts",20),bg="navy",fg="orange",activeforeground="navy",activebackground="orange",border=0)

b_resume.pack()

b_savedplay=Button(window,text="Countinue Playing",command=lambda: p.saved_play(),font=("small fonts",20),bg="navy",fg="orange",activeforeground="navy",activebackground="orange",border=0)

b_savedplay.pack()

global globnum,resume,new,saved#this global var are used to check for old/new player

globnum=0

resume=False

new=False

saved=False

#quotes games

quotes=["Games are the most elevated form of investigation. --Albert Einstein",

"The game itself is bigger than the winning.--Dejan Stojanovic",

"Games lubricate the body and mind.--Benjamin Franklin",

"We do not stop playing because we grow old. \nWe grow old because we stop playing.--Benjamin Franklin",

"Life is more fun if you play games.--Roald Dahl"]

no_quote=random.randint(0,4)

pygame.mixer.music.load("Tobu - Hope [NCS Release].mp3")

pygame.mixer.music.play(loops=2)

label_quote=tk.Label(window,text=quotes[no_quote],font=("Harrington",20),fg="orange",bg="navy").pack()

label_thank=tk.Label(window,text="|^_^|Hope You Have A Great Day|^_^|.",font=("small fonts",25),fg="orange",bg="navy").pack()

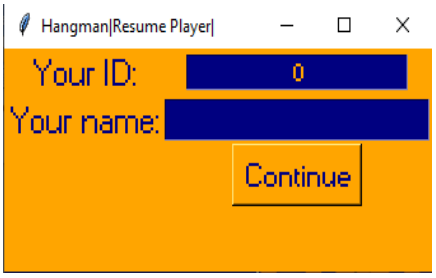
label_made=tk.Label(window,text="Made by : \n Sanskar Sharma and Devansh Talan",font=("small fonts",25),bg="navy",fg="orange").pack()

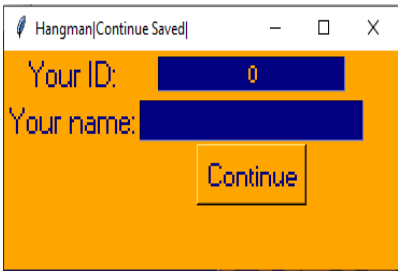
#----------Mainloop_ahead----------

window.mainloop()

#----------Dead_End----------

SAMPLE OUTPUTS






Hangman|Play area|
HANGMAN
Save!
_ _ _ _
Guessed Letters:
Enter


HANGMAN
Save!
_ _ ?_
Enter


Hangman(Play area)
HANGMAN
Save!
_°_°_
Letters Used:e,d,y,s,t,o,a,t,j,r,
Enter
HINT2-cards game with any no of cards
You Lost,Better Luck Next Time!joker was the word.
Next Round!


Hangman(Play area)
HANGMAN
Save!
crow
Letters Used:s,o,e,a,g,c,w,t,e,r,
Enter
HINT2-bird of genus corvus
Congratulation, you saved the hangman|^o^|The word was: crow
Next Round!

FUTURE ENHANCEMENTS

Some of the future enhancement are:-

- More number of levels.
- Leaderboard.
- Topic selection by the user.

Bibliography

- Computer science textbook (XII) by Sumita Arora
- www.flaticon.com
- NCS[No Copyright Sound]
- GeeksforGeeks
- Internet