**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **1** | **SYNOPSIS** |
| **2** | **INTRODUCTION TO PYTHON** |
| **3** | **HARDWARE SPECIFICATION** |
| **4** | **PROJECT DESCRIPTION** |
| **5** | **CODING** |
| **6** | **OUTPUT** |
| **7** | **SCOPE OF THE PROJECT** |
| **8** | **BIBILIOGRAPHY** |

**PROJECT SYNOPSIS**

The main objective of this game system which is developed in Python is to provide entertainment and test the thinking & problem solving capabilities of the personnel working with this system. With this system working personnel should solve complex problems to finish the program. This system enables its users input the solutions and returns whether the solution is valid or not after the evaluation. The operations which can be performed by the personnel are: Problem Solving, Critical Thinking, Inserting solutions.

Python 3.7 has been used in the project.

**INTRODUCTION TO PYTHON**

Python is an object-oriented programming (OOP). It was created by Gudio Van Rossum and released in 1991. It is used for developing desktop GUI application, website and web applications.

Advantages and Features of Python

* Simple: Python is a simple and minimalistic language. The pseudo- code nature of Python is one of its greatest strengths.
* Easy to Learn: Python has an extraordinarily simple syntax which makes it easier to learn.
* Free and Open Source: One can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.
* High –Level-Language: While writing programs in Python, one does not need to worry about low-level details such as managing the memory used by the program.
* Portable: Python can work on many platforms like Linux, Windows etc.
* Interpreted: Python does not need compilation to binary. The program is run directly from the source code.
* Extensible: A Python code can be written in C or C++ language and that can be compiled in C/C++ language.
* Embeddable: Python can be embedded within C/C++ to give scripting capabilities for the program’s users.
* Extensive Libraries: The Python Standard Library is a collection of script modules accessible to a Python program to simplify the programming process and removing the need to rewrite commonly used commands. They can be used by ‘calling/importing’ them at the beginning of a script.

**Major Python Modules**

Major Python modules used in the project are listed below:

**Sys Module:**

The sys module provides functions and variables used to manipulate different parts of the Python runtime environment.

**Mysql.connector Module:**

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

**Random Module**

Random module implements pseudo-random number generators for various distributions, including integer and float (real).

**Hashlib Module**

Hashlib module implements a common interface to many different secure hash and message digest algorithms. Included are the FIPS secure hash algorithms SHA1, SHA224, SHA256, SHA384, and SHA512 (defined in FIPS 180-2) as well as RSA’s MD5 algorithm.

.

**HARDWARE SPECIFICATION**

PROCESSOR : Intel(R) Core(TM) i3-4150 CPU

@ 3.50GHz

RAM : 4.00 GB (3.38 GB usable)

HARD DISK : ST3500312CS

CD ROM : ASUS DRW-24DMT

MONITOR : Dell E1916HV (Analog)

KEYBOARD : HID Keyboard Device

MOUSE : Logitech K120

**PROJECT DESCRIPTION**

This Game System can be maintained by anyone. This game system provides complex problems and most of the problems can only be solved by critical thinking. This game system can be used to evaluate the problem solving capabilities, level of observation and basic knowledge.

**Game System**

Game System will provide users a set of problems numbered as tasks. The main objective of the user is the guess out the correct number that is generated randomly within the range of 1-5. If the user cannot guess out correct number, the user will be provided with tasks to get a chance again to guess out the correct number. The randomly generated number will be same throughout the game. To get another chance to solve the problem the user should solve the complex problem. The user will get only one chance to a task.

If the solution of the task in invalid the user will be moved to the second task. At the end the user will be prompt to play **tictaktoe** with another user. Only failed users will reach this level and this is the ultimate end of the program.

**CODING**

**Attendance recorder**

#!/usr/bin/env python3

import random

import sys

import hashlib

def manual(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('----------------------------------TASK-7 MANUAL----------------------------------')

print('\n')

print('This is a multiplayer! So make sure that you have a friend with you to play with.')

print('You should enter numbers!(in Coordinates)')

print('Usage:')

print(' Enter the coordinates: > "number1"<space>"number2"')

print('Coordinates should be from 1 to 3!')

print('"number1" = column and "number2" = row')

print('Note! ROWS ARE COUNTED FROM BOTTOM')

task7(num\_task,user)

def print\_board(str\_list):

print('---------')

for index in range(0,3):

subline = str\_list[index\*3:index\*3+3]

print(f'| {" ".join(subline)} |')

print('---------')

#range

def three\_in\_row(line):

if line == 'XXX':

return 'X'

elif line == 'OOO':

return 'O'

else:

return None

def validate\_game(str\_list):

x\_count = 0

o\_count = 0

for l in str\_list:

if l == 'X':

x\_count += 1

if l == 'O':

o\_count += 1

if(abs(x\_count - o\_count ) >= 2):

return False

else:

return True

def judge(str\_list):

if not validate\_game(str\_list):

return 'Impossible'

sublines = []

x\_win = False

o\_win = False

result = None

for index in range(0,3):

sublines.append(''.join(str\_list[index\*3:index\*3+3]))

sublines.append(''.join([str\_list[index], str\_list[index + 3], str\_list[index + 6]]))

sublines.append(''.join([str\_list[0], str\_list[4], str\_list[8]]))

sublines.append(''.join([str\_list[2], str\_list[4], str\_list[6]]))

# loop all the possible rows

for line in sublines:

result = three\_in\_row(line)

if result == 'X':

x\_win = True

if result == 'O':

o\_win = True

if x\_win and o\_win:

return 'Impossible'

elif x\_win or o\_win:

return 'X wins' if x\_win else 'O wins'

elif '\_' in str\_list:

return 'Game not finished'

else:

return 'Draw'

def is\_int(val):

try:

\_ = int(val)

except ValueError:

return False

return True

def verify\_move(current, move\_input, player):

move = move\_input.split(' ')

for t in move:

if not is\_int(t):

return False, 'You should enter numbers!'

if int(t) > 3 or int(t) < 1:

return False, 'Coordinates should be from 1 to 3!'

# find current location

move\_int = [int(move[0]), int(move[1])]

loc = 3\* (3 - move\_int[1]) + move\_int[0] - 1

loc\_state = current[loc]

if loc\_state != '\_':

return False, 'This cell is occupied! Choose another one!'

else:

current[loc] = player

return True, current

def init\_game():

return list('\_\_\_\_\_\_\_\_\_')

def user\_input(game\_state, player):

valid\_move = False

while(not valid\_move):

move\_input = input("Enter the coordinates: > ")

valid,result = verify\_move(game\_state, move\_input, player)

if not valid:

print(result)

valid\_move = False

else:

game\_state = result

valid\_move = True

return game\_state

def game\_begins():

heading = "\*\* GAMBLING WITH THE SECRET NUMBER \*\*"

print("\*" \* len(heading))

print(heading)

print("\*" \* len(heading))

def game(num\_task,user):

global secret\_number

guess = int(input(user+" ,Please enter your guess: "))

if (guess == secret\_number):

for i in range(5):

print (user+" ,you are amazing..your guess is absolutely right")

elif(guess != secret\_number):

print (user+" ,Your guess is wrong")

print(user+' ,now its task time!')

if (num\_task==1):

task1(num\_task,user)

if(num\_task==2):

task2(num\_task,user)

if(num\_task==3):

task3(num\_task,user)

if(num\_task==4):

task4(num\_task,user)

if(num\_task==5):

task5(num\_task,user)

if(num\_task==6):

task6(num\_task,user)

if(num\_task==7):

manual(num\_task,user)

else:

sys.exit()

def game1(num\_task):

user = input('Enter your name:')

print(user+' ,let us start the game....')

global secret\_number

game(num\_task,user)

#TASK 1

def task1(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('TASK 1')

numbers = []

for charater in user:

numbers.append(ord(charater))

for number\_1 in numbers:

print(hex(number\_1))

print('Guess the hidden word')

word=input('>')

if (word == user):

print(user+" .YES, you have successfully completed the task ")

print('lets try again')

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task = num\_task+1

task2(num\_task,user)

##TASK 2

def task2(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('TASK 2')

a=input('Print the hidden word- (16,25,20,8,15,14):')

a1='python'

if (a == a1):

print(user+" ,YES the word is right, you have successfully completed the task ")

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task = num\_task+1

task3(num\_task,user)

#TASK 3

def task3(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('TASK 3')

num\_1 = int(input("Please Enter the Number to Check for Armstrong: "))

sum = 0

temp = num\_1

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

if num\_1 == sum:

print(user+" ,YES it is an armstrong no., you have successfully completed the task ")

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task = num\_task+1

task4(num\_task,user)

#TASK 4

def task4(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('TASK 4')

name\_hash = hashlib.sha256(user.encode('utf-8')).hexdigest()

print('Find out the hidden string!!')

print(name\_hash)

checker = input('>')

if(user == checker):

print('YES!! You have successfully completed the task')

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task = num\_task+1

task5(num\_task,user)

#Task 5

def task5(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('Task 5')

rows = 8

for i in range(0, rows):

for j in range(0, i + 1):

print("\*")

print(" ")

print('Can you see an amazing pattern?')

a=int(input('Enter the number of rows in this pattern: '))

if (a == rows):

print(user+' ,yes,you have successfully completed the task')

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task = num\_task+1

task6(num\_task,user)

#Task 6

def task6(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('Task 6')

print('who created python?')

a='Monty Python'

b='Guido van Rossum'

c='Kaustubh'

print('option 1=',a)

print('option 2=',b)

print('option 3=',c)

ur\_ans = int(input('enter an option(1/2/3):'))

org\_ans = 2

if(ur\_ans == org\_ans):

print('Excellent!!you even knew this!!,you have successfully completed the task' +user)

game(num\_task,user)

else:

print('Task failed, better luck next time.')

num\_task=num\_task+1

manual(num\_task,user)

#task7

def task7(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('TASK 7')

#print(secret\_number)

game\_over = False

players = ['X', 'O']

rounds = 0

game\_state = init\_game()

while(not game\_over):

print\_board(game\_state)

game\_state = user\_input(game\_state, players[rounds % 2])

out = judge(game\_state)

rounds += 1

if 'wins' in out or out == 'Draw':

print\_board(game\_state)

print(out)

game\_over = True

restart\_task7=input('Do you want to play again this task?(Y/n):')

if restart\_task7.lower == 'y':

manual(num\_task,user)

task8(num\_task,user)

#Task 8

def task8(num\_task,user):

global secret\_number

print('\n')

print('----------------------------------')

print('\n')

print('Thank you '+user)

print("Aren't you curious to know the answer?" )

print(secret\_number,'is the mysterious secret number')

print('Good bye '+user)

task9(num\_task,user)

#end of the program

def task9(num\_task,user):

print('\n')

print('----------------------------------')

print('\n')

print('exiting game...')

sys.exit()

num\_task = 1

game\_begins()

secret\_number = random.randrange(1, 5)

game1(num\_task)

**OUTPUT**

Connected to DB

Ready to uses...

1- Login

2- New Account

3- Exit

> 1

**Login:**

1- Login

2- New Account

3- Exit

> 1

Username: \*\*\*\*\*\*\*\*\*\*

Password: \*\*\*\*\*\*\*\*\*\*

1) Delete your account

2) Update password

3) Exit

**Deletion of account:**

1- Login

2- New Account

3- Exit

> 1

Username: \*\*\*\*\*\*\*\*\*\*

Password: \*\*\*\*\*\*\*\*\*\*

1) Delete your account

2) Update password

3) Exit

> 1

Are You Sure You Want To Delete?(Y/n): Y

Your account is Deleted..

**Password updating:**

1- Login

2- New Account

3- Exit

> 1

Username: \*\*\*\*\*\*\*\*\*\*

Password: \*\*\*\*\*\*\*\*\*\*

1) Delete your account

2) Update password

3) Exit

>2

New Password: \*\*\*\*

Confirm Password: \*\*\*\*

**New Account:**

1- Login

2- New Account

3- Exit

> 2

Name: Testaccount

Username: test123

Password: \*password

Email: [test@institution.com](mailto:test@institution.com)

Successfully created!

**Admin Access:**

1- Login

2- New Account

3- Exit

>945

Name Username Email Password

User1 User123 [user1@mail.com](mailto:user1@mail.com) dee4164777a98291e138fcebcf7ea59a837226bc8388cd1cf694581586910a81d46f07b93c068f17eae5a8337201af7d51b3a888a6db41915d801cb15b6058e5

Testaccount test123 [test@institution.com](mailto:test@institution.com) dee4164777a98291e138fcebcf7ea59a837226bc8388cd1cf694581586910a81d46f07b93c068f17eae5a8337201af7d51b3a888a6db41915d801cb15b6058e5

Username Time

User123 2021, 1, 31, 15, 30, 40, 457057

**SCOPE OF THE PROJECT**

* To test the thinking capabilities of the user.
* To test the problem solving skill of the user.

**BIBILIOGRAPHY**

COMPUTER SCIENCE WITH Python XI – Sumita Arora

COMPUTER SCIENCE WITH Python XII– Sumita Arora

COMPUTER SCIENCE WITH Python XII – Preeti Arora

[https://www.python.org](https://www.python.org/)