

A Real Time Research Project Report on

THE HANGMAN GAME

Submitted to the

Guru Nanak Institute of Technology

(Autonomous)

In partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

By

KALUSANI SANTHOSH REDDY(22831A6649)

GOWNI PRASHANTH REDDY(22831A6636)

B ANVITH RAJ(22831A6606)

Under the Esteemed Guidance of

Mr. G. Manoj

Assistant Professor

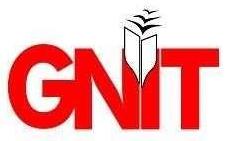
Department of Emerging Technologies



Department of Emerging Technologies

**Guru Nanak Institute of Technology
(Autonomous)**

Ibrahimpatnam, R.R District- 501506



GURU NANAK INSTITUTE OF TECHNOLOGY
(Autonomous)
Ranga Reddy District -501506

Department of Emerging Technologies

CERTIFICATE

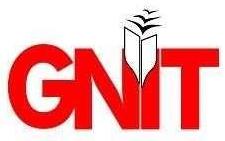
This is to certify that Real Time Research Project report entitled "**THE HANGMAN GAME**" by **KALUSANI SANTHOSH REDDY (22831A6649), GOWNI PRASHANTH REDDY (22831A6636), B ANVITH RAJ (22831A6606)**,

was submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) of Guru Nanak Institute of Technology during the academic year 2023-2024 is a Bonafide record of work carried out under our guidance and supervision.

Mr. G. MANOJ
INTERNAL GUIDE

Mr. B. PRANEETH REDDY
PROJECT CO-ORDINATOR

Dr. M. SADISH SENDIL
HOD-ET



VISION AND MISSION

VISION OF GNIT

To be a world –class educational and research institution in the service of humanity by promoting high quality Engineering and Management Education

MISSION OF GNIT

M1: Imbibe soft skills and technical skills.

M2: Develop the faculty to reach the international standards.

M3: Maintain high academic standards and teaching quality that promotes the analytical thinking and independent judgment.

M4: Promote research, innovation, and product development by collaboration with reputed foreign universities.

M5: Offer collaborative industry programs in emerging areas and spirit of enterprise.



DEPARTMENT OF EMERGING TECHNOLOGIES

VISION OF DEPARTMENT

To be a premier Department of Emerging Technologies, known globally for value based education, Research & Innovation through academic excellence to serve the needs of industry and society

MISSION OF DEPARTMENT

M1: Develop a strong foundation on fundamentals of Computer Science and Engineering through outcome based Teaching Learning process.

M2: Establish state of art & Centre of Excellence facilities in Emerging technologies for Design and development of Innovative Products.

M3: Involve the students in Group activities, including those of professional bodies to develop leadership, entrepreneurship and good communication skills.



DEPARTMENT OF EMERGING TECHNOLOGIES

Program Educational Objectives (PEO's)

PEO-1: Have an ability to adapt to a rapidly changing environment by having learned and applied cutting-edge skills in emerging technologies.

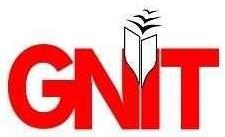
PEO-2: Build a strong foundation in the field of artificial intelligence, cyber security, machine learning, and data science to contribute significantly in the area of innovations and research.

PEO-3: Have strong work ethics, professional attitude, team spirit, leadership, and enterprising skills to serve industry and society.

Program Outcomes (PO's)

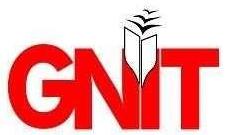
1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



MAPPING WITH PO'S AND PEO'S

Sl. No.	Content	PO's	PEO's
1	Introduction	PO1, PO2	PEO1
2	Abstract	PO1, PO2	PEO1
3	System Specifications	PO3, PO5	PEO1
4	Architecture Design	PO3, PO4	PEO1, PEO2
5	Implementation	P09, PO10, PO11	PEO2, PEO3
6	Results	PO4, PO11, PO12	PEO2, PEO3



DECLARATION

We hereby declare that the Real time research project report entitled "**THE HANGMAN GAME**" is the work done by **KALUSANI SANTHOSH REDDY (22831A6649)**, **GOWNI PRASHANTH REDDY (22831A6636)**, **B ANVITH RAJ (22831A6606)**, towards the fulfilment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and Engineering (Artificial intelligence and Machine Learning), Guru Nanak Institute of Technology, is the result of the work carried out under the guidance **Mr. G.MANOJ**, Assistant Professor , Guru Nanak Institute of Technology, Hyderabad.

We further declare that this real time project report has not been previously submitted before either in part or full for the award of any degree or any diploma by any organization or any universities

KALUSANI SANTHOSH REDDY(22831A6649)
GOWNI PRASHANTH REDDY(22831A6636)
B ANVITH RAJ(22831A6606)

ACKNOWLEDGEMENT

“Task Successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it happen.

We would like to express our sincere thanks and gratitude to our principal **Dr.S. SREENATHA REDDY** and Head of the Department **Dr. M. SADISH SENDIL**, Department of Emerging Technologies, Guru Nanak Institute of Technology for guided us in developing the requisite capabilities for taking up this mini project.

We would like to say our sincere thanks to **Mr. B.PRANEETH REDDY** Assistant Professor and Project Coordinator, Department of CSE (AIML), GNIT for providing seamless support and right suggestions are given in the development of the project.

We would like to say our sincere thanks to **Mr. G. MANOJ, Assistant Professor** and Project Guide, Department of CSE (AIML), GNIT, for providing seamless support and right suggestions are given in the development of the project.

Finally, we would like to thank our family members for their moral support and encouragement to achieve their goals.

KALUSANI SANTHOSH REDDY(22831A6649)

GOWNI PRASHANTH REDDY(22831A6636)

B ANVITH RAJ(22831A6606)

THE HANGMAN GAME

ABSTRACT

The project uses Python programming language to randomly select a secret word and allow a player to guess letters with limited chances to win. If a guessed letter appears in the word, all instances are revealed. If not, the player loses a chance. The goal is to guess the word before running out of chances.

This is a simple Hangman game using Python programming language. We can use this as a small project to boost their programming skills and understanding logic. The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it. Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins.

TABLE OF CONTENTS

ABSTRACT:	xi
TABLE OF CONTENTS	xii
TABLE OF FIGURES	xiii
CHAPTER-1 INTRODUCTION	1
1.1 General	1
1.2 SCOPE OF THE PROJECT	2
1.3 OBJECTIVE	3
1.4 EXISTING SYSTEM:	4
1.4.1 METHODOLOGY:	4
1.5 LITERATURE SURVEY	6
1.6 PROPOSED SYSTEM	7
CHAPTER 2 PROJECT DESCRIPTION	8
2.1 GENERAL	8
2.2 DEFINITION OF GAMES.....	9
2.3 TECHNIQUE USED OR ALGORITHM USED.....	10
CHAPTER-3 REQUIREMENTS	12
3.1 HARDWARE REQUIREMENTS	12
3.2 SOFTWARE REQUIREMENTS	13
CHAPTER 4 DEVELOPMENT TOOLS	14
4.1 GENERAL	14
4.2 HISTORY OF PYTHON	14
4.3 IMPORTANCE OF PYTHON	14
4.4 FEATURES OF PYTHON	15
4.5 LIBRARIES USED IN PYTHON	15

CHAPTER 5 IMPLEMENTATION	17
5.1. CODE IMPLEMENTATION	17
5.2. CODE EXPLANATION.....	20
5.3 ILLUSTRATION	21
5.4 GITHUB REPOSITORY	21
CHAPTER 6 SNAPSHOTS	22
6.1 HOW I PLAY.....	22
CHAPTER 7 FUTURE ENHANCEMENTS	24
CHAPTER 8 CONCLUSION	25

LIST OF FIGURES:

DATA FLOW DIAGRAM.....	21
IN GAME PROCESS.....	22,23

CHAPTER-1 INTRODUCTION

1.1 GENERAL

Python is one of the many open source object oriented programming application software available in the market .Python is developed Guido van Rossum. Guido van Rossum started implementing Python in 1989. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues .Some of the many uses of Python are application development, implementation of automation testing process, allows multiple programming build, fully constructed programming library, can be used in all the major operating systems and platforms, database system accessibility, simple and readable code, easy to apply on complex software development processes, aids in test driven software application development approach, machine learning/ data analytics, helps pattern recognitions, supported in multiple tools, permitted by many of the provisioned frameworks, etc.

The project uses Python programming language to randomly select a secret word and allow a player to guess letters with limited chances to win. If a guessed letter appears in the word, all instances are revealed. If not, the player loses a chance. The goal is to guess the word before running out of chances.

1.2 SCOPE OF THE PROJECT

Hangman is used often by teachers to practice spelling, vocabulary and just for fun. The most popular way to play hangman games offline is to draw blank letters for the chosen word on a paper or on the blackboard and let the players guess the letters.

Problem solving skills are developed by playing this game because the players must try to guess and figure out what word their opponent is trying to spell based on the length of the word and the letter they can guess to fill in.

1.3 OBJECTIVE

The project uses Python programming language to randomly select a secret word and allow a player to guess letters with limited chances to win. If a guessed letter appears in the word, all instances are revealed. If not, the player loses a chance. The goal is to guess the word before running out of chances.

The objective of the game Hangman is to guess the word (phrase) in the time limit. Bonus for guessing a word' will be added to your score for guessing a word. This is usually a game for two players but can be played with any number of guessers.

1.4 EXISTING SYSTEM:

The main part of the Hangman program displays the name of the game, sets up some variables, and executes a while loop. This section walks through. welcome= ['Welcome to Hangman! A word will be chosen at random and', 'you must try to guess the word correctly letter by letter', 'before you. Python Code for Hangman Game.

The object of hangman is to guess the secret word before the stick figure is hung. Players take turns selecting letters to narrow the word down. Players can take turns or work together. Gameplay continues until the players guess the word or they run out of guesses and the stick figure is hung.

1.4.1 METHODOLOGY:

The word to guess is represented by a row of dashes, giving the number of letters. If the guessing player suggests a letter which occurs in the word, the program writes it in all its correct positions. One player thinks of a word and the other tries to guess it by suggesting the letters.

1.5 LITERATURE SURVEY

Title: Hangman is a guessing game for two or more players. One player thinks of a word, phrase, or sentence and the other(s) tries to guess it by suggesting letters or numbers within a certain number of guesses. Originally a paper-and-pencil game, there are now electronic versions.

Author: Alice Gomme

Year: 1894

Description:

The word to guess is represented by a row of dashes representing each letter or number of the word. Rules may permit or forbid proper nouns, such as names, places, brands, or slang. If the guessing player suggests a letter which occurs in the word, the other player writes it in all its correct positions. If the suggested letter does not occur in the word, the other player adds (or alternatively, removes) one element of a hanged stick figure as a tally mark. Generally, the game ends once the word is guessed, or if the stick figure is complete — signifying that all guesses have been used.

The player guessing the word may, at any time, attempt to guess the whole word. If the word is correct, the game is over and the guesser wins. Otherwise, the other player may choose to penalize the guesser by adding an element to the diagram. If the guesser makes enough incorrect guesses to allow the other player to complete the diagram, the guesser loses. However, the guesser can also win by guessing all the letters that appear in the word, thereby completing the word, before the diagram is completed.

Title: Through Hangman game, the students able to place the words in a sentence orderly, they able to spell English word fluently also identify the meaning of words. Through Hangman game, the students were able to recollect the vocabularies that had been taught before

Author: Karen King-Aribisala's

Year: 2007

Description:

- The object of hangman is to guess the secret word before the stick figure is hung. Players take turns selecting letters to narrow the word down.
- Players can take turns or work together. Gameplay continues until the players guess the word or they run out of guesses and the stick figure is hung.
- If you want to play with younger kids, use a snowman instead of a hangman to avoid scaring or often anyone.

1.6 PROPOSED SYSTEM

One player thinks of a word and the other tries to guess it by suggesting the letters. The word to guess is represented by a row of dashes, giving the number of letters. If the guessing player suggests a letter which occurs in the word, the program writes it in all its correct positions.

1.6.1 PROPOSED SYSTEM ADVANTAGES

The players guess letter by letter, while the host draw the part of hangman. This game focus on spelling, pronunciation and vocabulary. Wiratania (2018), Hangman game can be used to enlarge vocabulary mastery, practice spelling, and trained students' concentration. This game helps teacher in controlling classroom.

CHAPTER 2 PROJECT DESCRIPTION

2.1 GENERAL

According to Rabin (2005), game can be divided into several types. Game of the same kind of elements, visual style and method of playing can be group into the same category. Computer games can be divided into different categories including action, role-playing, strategy, stimulation and puzzle.

Word games are generally engaged as a source of entertainment, but have been found to serve an educational purpose as well. Word games and puzzles are spoken or board games often designed to test ability with language or to explore its properties. For instance, young children can find enjoyment playing modestly competitive games such as Hangman, while naturally developing important language skills like spelling. Solving crossword puzzles, which requires familiarity with a larger vocabulary, is a pastime that mature adults have long credited with keeping their minds sharp.

Word games take many different forms. For example, letter arrangement games ask the player to form words out of a seemingly random string of letters. Examples of letter arrangement word games include Scrabble, Boggle, Up words, and Literati.

2.2 DEFINATION OF GAMES

Structured games are word games that focus on the semantics of words. In these games, players are asked to interpret word, picture, or action clues to solve the puzzle. Examples of structured word games include Charades, Taboo, and Startegies.

The potential for play is another key dramatic element that engages players emotionally in games. In games, the constraints of the rules and procedures are the rigid structure, and the play within that structure is the freedom of players to act within those rules. Play helps us learn skills and acquire knowledge, socialize, assists us in problem solving, allows us to relax and makes us see things differently. Play as a process of experimentation is an area of common ground for artists, scientists as well as children. Play is recognized as a way of achieving innovation and creativity because it helps us see things differently or achieve unexpected results.

2.3 TECHNIQUES AND ALGORITHMS USED

Suppose we are playing the game [hangman](#). My opponent and I both have access to the dictionary during the game. My opponent picks a word from the dictionary with knowledge of the algorithm which I will use to guess his secret word. Once my opponent has picked a word, I am given only the length of the word. I can guess a letter which I think will be in the word, and if it is in the word my opponent identifies all of the positions of that letter in that word. If I guess a letter which isn't in the word, that's counted as one mistake. If I can guess the word before too many mistakes I win.

My opponent's goal should be to pick a word which maximizes the number of mistakes (incorrect guesses) I make before I can guess the word. My goal is to minimize them. (Traditionally, if the number of mistakes is $>$ some number then I lose the game, but we can relax that constraint.)

Consider three algorithms for letter guessing. These are the main ones I've thought of, but there are probably more, and I welcome alternate ideas. In all three algorithms, the first step will be to gather the list of words which are the same length as the secret word. Then, for each letter in the alphabet, count the number of words in the dictionary which contain that letter. For example, maybe 5000 contain "a", 300 contain "b", and so on. Here it is in python:

CODE:

```
alphabet = set('abcdefghijklmnopqrstuvwxyz')
probabilities = {letter:0 for letter in alphabet}

for word in dictionary:
    for letter in word:
        if letter in alphabet:
            probabilities[letter] += 1

# now we have the letter frequencies
```

THE THREE ALGORITHMS DIVERGE:

1. In the first algorithm, we guess the letter which the most number of remaining words contain. So if 5000 remaining words contain "a" and no other letters are in that many words, we will pick "a" every time. If "a" is correct, this gives us positional information which we can filter the list further. For example, we might filter the list by all words that match ".a.". (Dots are unknown letters.) If "a" is incorrect, we filter out all words which contain "a". In the case where there is a tie and two letters are found in an equal number of words, letters are chosen alphabetically. So if we know the word matches ".ays", we'll just guess words in alphabetical order.
2. This is only slightly different from the first algorithm. Instead of always choosing alphabetical ordering, in the case of a tie we choose letters randomly. This has the benefit that our opponent doesn't know what we will pick. In the first strategy, "rays" is always better than "days". This avoids that issue.
3. In this case, we pick letters with a probability proportional to the number of words which contain that letter. At the beginning when we tallied the number of words which contain "a" and the number which contain "b" and so on, since "a" happened to be found in the most number of words, in strategies 1 and 2 we picked "a" 100% of the time. Instead, we will still choose "a" a plurality of the time, but a small number of times we'll pick "z" even though it might be found in 10x more words. I have my doubts about this strategy being optimal but [it was used in research in 2010](#)

CHAPTER 3 REQUIREMENTS

These are the requirements for doing the project. Without using these tools & software's we cannot do the project. Therefore, we have two requirements to do the project.

They are

- Hardware Requirements
- Software Requirement

3.1 HARDWARE REQUIREMENTS:

RAM: 4GB

Disk Space: 1TB

Operating System: Windows

Processor: Intel i5

3.2 SOFTWARE REQUIREMENTS:

Python-3.10.0 :

- Using its various extensive libraries ,modules ,lists and dictionaries we create the actual core code of our game using its object oriented programming and GUI features we will create our source code

HTML:

- using hypertext markup language and cascading style sheets we will develop the structure and interface of our project

CSS:

- Cascading Style sheets are helpful for adding styles ,background ,font for a user-appealing look of our project

CHAPTER 4 DEVELOPMENT TOOLS

4.1 GENERAL

The development of our project involves various tools and technologies to ensure robust functionality and ease of use. Key development tools include the Python programming language and its extensive libraries, as well as the gymnasium, ipython, mathplotlib, numpy, etc library for multiple purposes. These tools are chosen for their simplicity, efficiency, and wide adoption in the development community, enabling developers to quickly build and deploy the application across multiple platforms.

4.2 HISTORY OF PYTHON

Python, created by Guido van Rossum and first released in 1991, is a high-level, interpreted programming language known for its readability and simplicity. Python's development was guided by the philosophy of emphasizing code readability and simplicity, allowing developers to express concepts in fewer lines of code compared to languages such as C++ or Java. Over the years, Python has evolved with contributions from the open-source community, resulting in a versatile language used in various domains including web development, data analysis, artificial intelligence, scientific computing, and more.

4.3 IMPORTANCE OF PYTHON

Python has become one of the most important programming languages in the world due to its simplicity, versatility, and the powerful standard library that supports a wide range of applications. Its readability and straightforward syntax make it an excellent choice for beginners and experienced developers alike. Python's importance is further highlighted by its extensive use in educational institutions, research projects, and industry applications. Its ability to integrate with other languages and tools, coupled with a large community that continuously contributes to its development, ensures Python remains a vital tool in the developer's arsenal.

4.4 FEATURES OF PYTHON

Python provides a multitude of features that contribute to its popularity:

- **Easy to Learn and Use:** Python's syntax is simple and clear, making it accessible for beginners.
- **Interpreted Language:** Python code is executed line-by-line, which makes debugging easier.
- **Versatile and Extensible:** Python can be used for various types of programming, including web development, data science, automation, and more.
- **Large Standard Library:** Python's extensive standard library provides modules and functions for many common tasks, reducing the need for external libraries.
- **Open Source:** Python is freely available and open source, allowing anyone to use, modify, and distribute it.
- **Cross-Platform:** Python runs on various operating systems including Windows, macOS, and Linux, making it a cross-platform language.
- **Strong Community Support:** A large and active community ensures continuous improvement and extensive resources for learning and problem-solving.

4.5 LIBRARIES USED IN PYTHON

In the development of the AI car game, several key Python libraries are utilized:

Gymnasium:

- Functionality: Gymnasium is a toolkit for developing and comparing reinforcement learning algorithms. It provides a standard API to communicate between learning agents and environments.

IPython:

- Functionality: IPython provides an enhanced interactive Python shell with powerful editing and debugging capabilities. It is often used for testing code snippets and quick iterations.

Matplotlib:

- Functionality: Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python.

NumPy:

- Functionality NumPy is a fundamental library for numerical computations in Python, providing support for arrays, matrices, and high-level mathematical functions:.

Pygame:

- Functionality: Pygame is a set of Python modules designed for writing video games. It includes computer graphics and sound libraries.

Torch:

- Functionality: Torch (PyTorch) is an open-source machine learning library used for applications such as computer vision and natural language processing. It provides tools for building and training neural networks. These libraries collectively enable the creation of a functional, user-friendly, car game leveraging Python's capabilities to deliver a robust application.

CHAPTER 5 IMPLEMENTATION

5.1 CODE IMPLEMENTATION:

```
# Python Program to illustrate
# Hangman Game
import random
from collections import Counter

someWords = '''apple banana mango strawberry
orange grape pineapple apricot lemon coconut watermelon
cherry papaya berry peach lychee muskmelon'''

someWords = someWords.split(' ')
# randomly choose a secret word from our "someWords" LIST.
word = random.choice(someWords)

if __name__ == '__main__':
    print('Guess the word! HINT: word is a name of a fruit')

    for i in word:
        # For printing the empty spaces for letters of the word
        print('_', end=' ')
    print()

    playing = True
    # list for storing the letters guessed by the player
    letterGuessed = ''
    chances = len(word) + 2
    correct = 0
    flag = 0
    try:
        while (chances != 0) and (flag == 0): # flag is updated when the word is correctly guessed
```

```

print()
chances -= 1

try:
    guess = str(input('Enter a letter to guess: '))
except:
    print('Enter only a letter!')
    continue

# Validation of the guess
if not guess.isalpha():
    print('Enter only a LETTER')
    continue
elif len(guess) &gt; 1:
    print('Enter only a SINGLE letter')
    continue
elif guess in letterGuessed:
    print('You have already guessed that letter')
    continue

# If letter is guessed correctly
if guess in word:
    # k stores the number of times the guessed letter occurs in the word
    k = word.count(guess)
    for _ in range(k):
        letterGuessed += guess # The guess letter is added as many times as it
occurs

# Print the word
for char in word:
    if char in letterGuessed and (Counter(letterGuessed) != Counter(word)):
        print(char, end=' ')
        correct += 1
# If user has guessed all the letters

```

```

# Once the correct word is guessed fully,
elif (Counter(letterGuessed) == Counter(word)):

# the game ends, even if chances remain

    print("The word is: ", end=' ')
    print(word)
    flag = 1
    print('Congratulations, You won!')
    break # To break out of the for loop
    break # To break out of the while loop

else:
    print('_', end=' ')

# If user has used all of his chances

if chances &lt;= 0 and (Counter(letterGuessed) != Counter(word)):

    print()
    print('You lost! Try again..')
    print('The word was {}'.format(word))

except KeyboardInterrupt:

    print()
    print('Bye! Try again.')
    exit()

```

IMPLEMENTATION:

This is a simple Hangman game using Python programming language. Beginners can use this as a small project to boost their programming skills and understanding logic.

The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it.

The Game: Here, a random word (a fruit name) is picked up from our collection and the player gets limited chances to win the game.

When a letter in that word is guessed correctly, that letter position in the word is made visible. In this way, all letters of the word are to be guessed before all the chances are over.

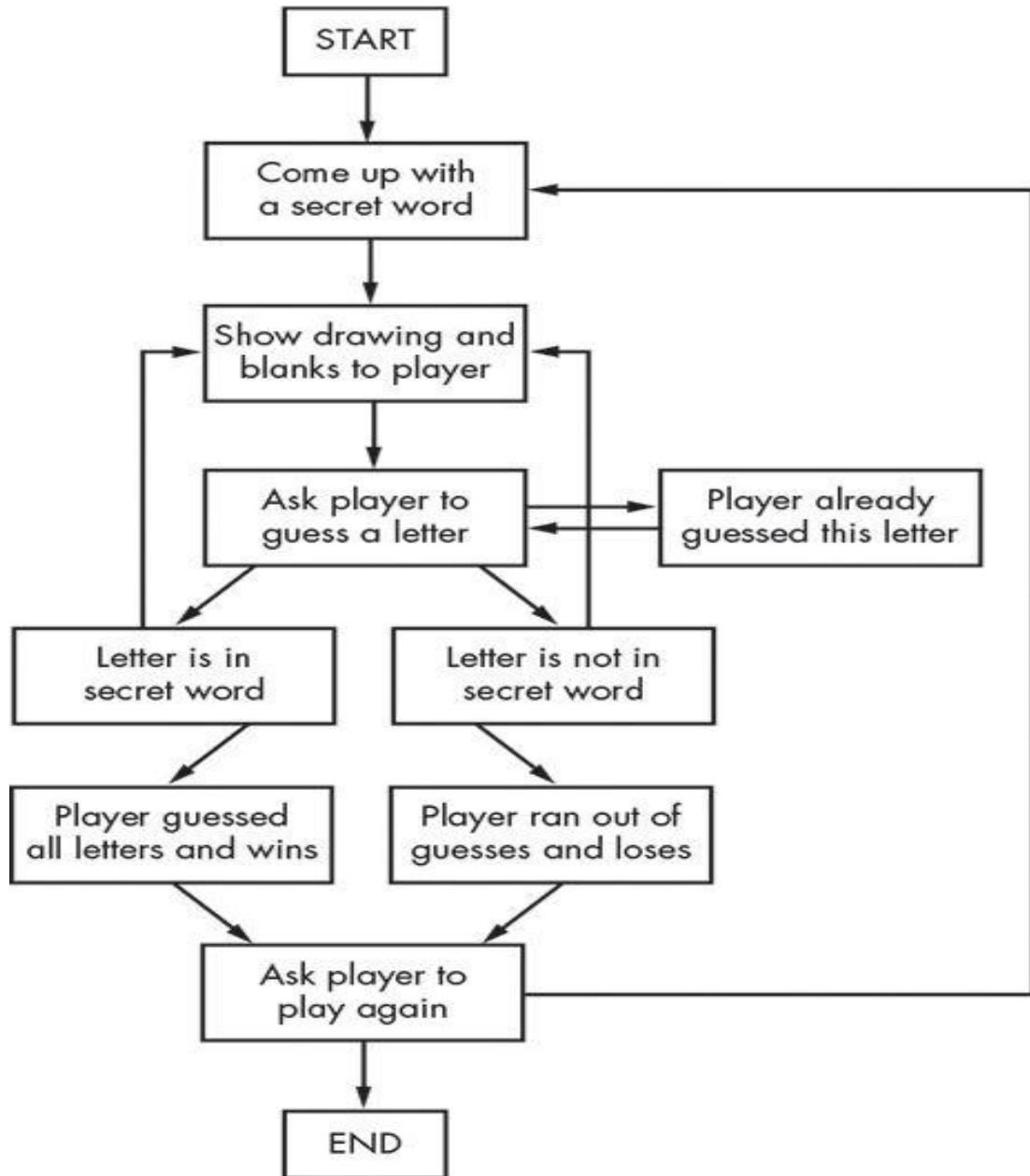
For convenience, we have given length of word + 2 chances. For example, word to be guessed is mango, then user gets $5 + 2 = 7$ chances, as mango is a five-letter word.

5.2 CODE EXPLANATION:

1. The code starts by importing the random module.
2. This module provides a way to generate random numbers.
3. Next, the code creates someWords, which is a list of fruit names.
4. The list is split into spaces using the string ‘ ‘, and then each space is replaced with a letter.
5. Next, the code randomly selects a secret word from our someWords list.
6. This word will be used as the input for the game later on.
7. The next part of the code checks to see if the user has entered an alpha character (a letter that appears in front of other letters).
8. If not, then they are asked to enter only a letter.
9. If they enter an alpha character, then it’s assumed that they want to guess at another letter in word .
10. So, this part of the code checks to see if guess matches one of the letters in word .

11. If it does, then chances is updated and flag is set to 1 .
12. Otherwise, chances is decreased by 1 and flag remains at 0 .
13. The next part of the code tries to guess at another letter in word .
14. If guess isn't valid (i.e., it doesn't match any of the letters in word), then print() prints out all empty spaces for letters in word , and
15. The code starts by importing the random module.
16. This module provides us with a number of useful functions, one of which is the choice function.
17. This function allows us to randomly choose a secret word from our list of words.
18. Next, we create some variables which will be used throughout the program.
19. These include someWords , word and letterGuessed .
20. letterGuessed will store the letter guessed by the player, while chances will store the number of times that the player has correctly guessed the word so far.
21. correct will keep track of how many letters have been guessed so far and flag will indicate whether or not the player has guessed the word correctly.
22. We then start looping through our list of words and randomly choosing a secret word from it.

5.3 ILLUSTRATION



5.4 GITHUB REPOSITORY

```
<script src="https://gist.github.com/lupinetti/8f89e5f33750aa7c91c3.js"></script>
```

CHAPTER 6 SNAPSHOT

HOW I PLAY ?

The screenshot shows the homepage of the "HangMan game". At the top, there is a navigation bar with "Home", "Play", and "Invite your friend" links. The main title "HangMan game" is displayed in a large, bold, blue font. Below the title, a banner with the text "Invite your friends to play!" is visible. In the center, there is a "Play Now" button and an "Invite your friend" button. To the left of these buttons, there is a "How I play?" section containing a text description and a small diagram of a hangman figure. To the right, there is a word representation consisting of a vertical line and several dashes.

How I play?

If the letter is contained in the word/phrase, the group or individual takes another turn guessing a letter. ... If the letter is not contained in the word/phrase, click the Play Again button, a portion of the hangman is added.

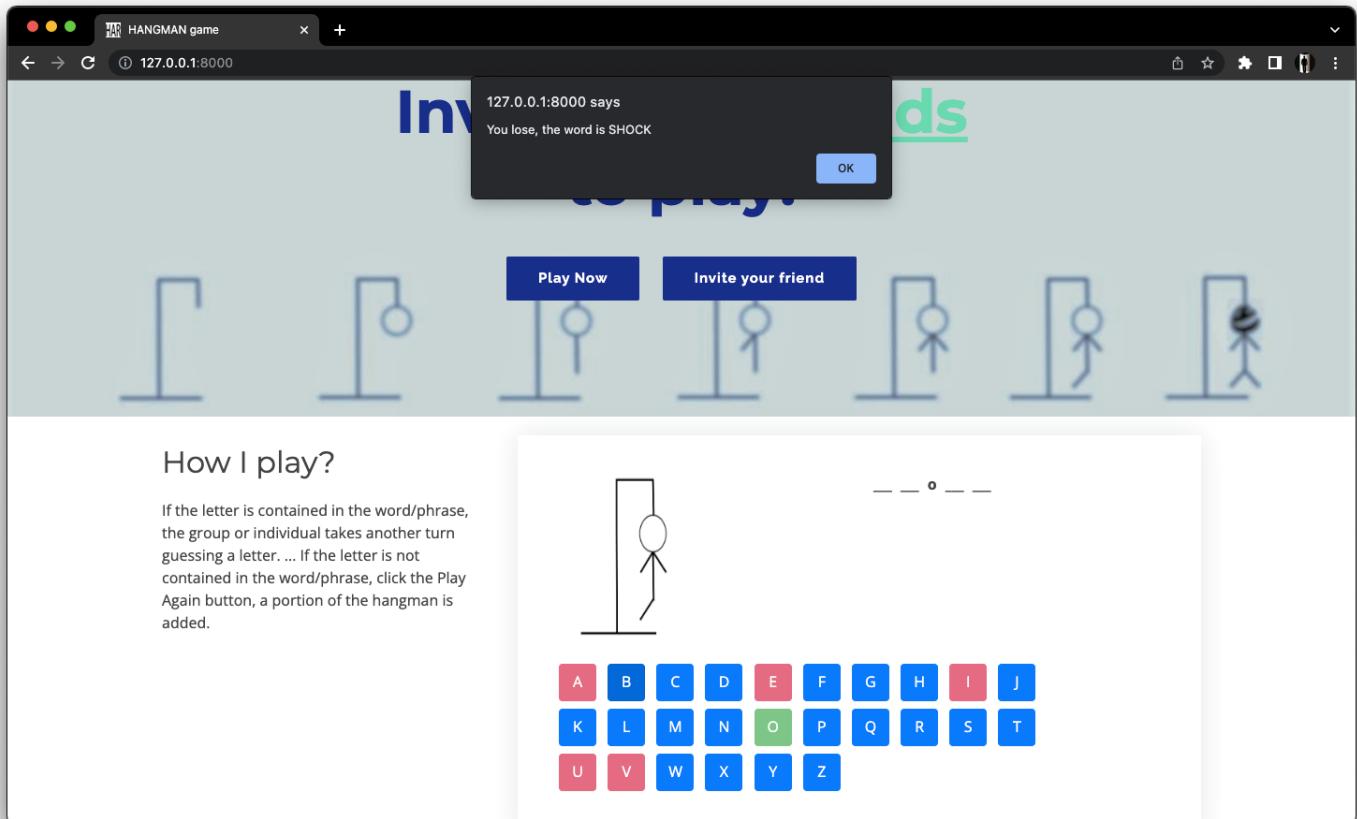
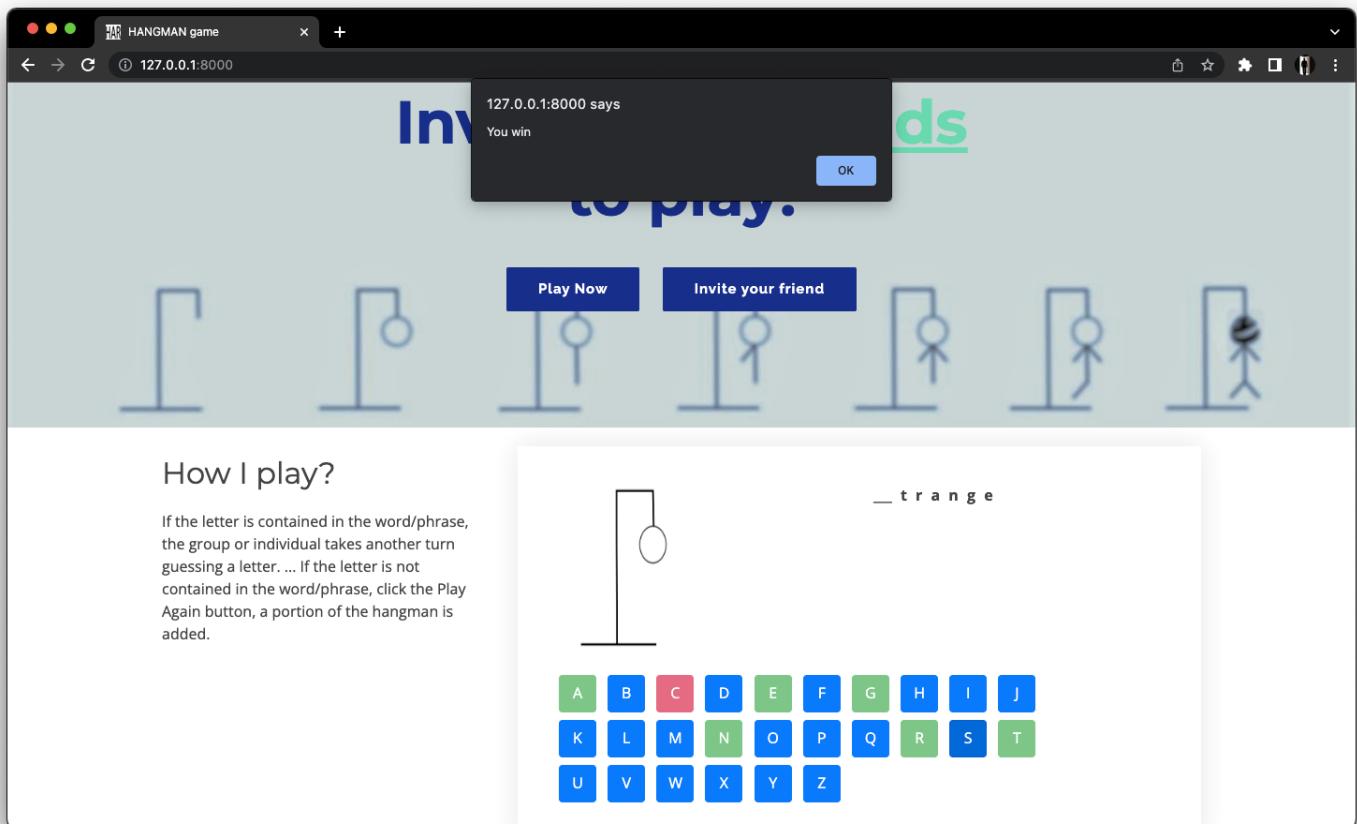
The screenshot shows the homepage of the "HangMan game". At the top, there is a navigation bar with "Home", "Play", and "Invite your friend" links. The main title "HangMan game" is displayed in a large, bold, blue font. Below the title, a banner with the text "Invite your friends to play!" is visible. In the center, there is a "Play Now" button and an "Invite your friend" button. To the left of these buttons, there is a "How I play?" section containing a text description and a small diagram of a hangman figure. To the right, there is a word representation consisting of a vertical line and several dashes. Below this, there is a grid of letters from A to Z. At the bottom, there is a "Generate new word" button and the word "always".

How can my friend play?

Just pick a random word, of our database, copy the url for share and voilá!!

Url for share <http://127.0.0.1:8000/fa2095e3-f122-4cce-be67-afe39c54cd68>

Dont forget ask to your friend a screenshot!!



CHAPTER 7 FUTURE ENHANCEMENTS

The Hangman game consists of the following components:

- words : A list of words from which the game selects a random word to be guessed.
- guesses : The number of remaining guesses.
- guessed_letters : A list of letters that have been guessed.
- root : The main window of the game.

Hangman is a guessing game for two or more players. One player thinks of a word, phrase, or sentence and the other(s) tries to guess it by suggesting . hangman: the main function that starts the game, handles user interaction, and creates and updates the figure.

CHAPTER 8 CONCLUSION

Generally, the game ends once the word is guessed, or if the stick figure is complete — signifying that all guesses have been used. The player guessing the word may, at any time, attempt to guess the whole word. If the word is correct, the game is over and the guesser wins.

A great deal was learned from the process of coding the hangman game, especially when it comes to problem solving. First, what worked best when trying to fix a bug in the code, it is crucial to first identify the source of the bug. This can be difficult sometimes when there are many lines of code, but if the problem is to be solved as fast as possible, this is the first thing to be done. This principle can also be applied to many other aspects of engineering as well. For example, when working on a manufacturing line and whatever product is being produced seems to be coming out with defects, there is a point in the line that this defect can be traced back to. When it comes to the process of coding itself, this project helped determine how important having structured pseudocode or a flowchart really was. One does not have to have these documents specifically written down somewhere, but should at least have a plan of their algorithm and a rough picture of what the code will look like in mind. This was how this hangman game was created. When it comes to further developing this game, it could still be made a lot more user-friendly. This could be done by having the program interact more with the user in a variety of ways. One specific way would be by including a “Play again?” screen, or even by providing instructions on how to enter guesses and what buttons to press.