NAME OF THE STUDENT: DHANASHRI CHAKRABORTY

Title: Hangman Game Using Python
Internship Task: Task 1 – Hangman

---

1. Introduction

This project is a simple text-based Hangman game developed using Python.
The goal of the game is to guess a hidden word one letter at a time.
The player has six chances to make incorrect guesses.

---

2. Objective

- To understand the basics of Python programming
- To apply loops, conditionals, lists, and string handling
- To build a small console-based game

---

3. Tools & Technology

- Python 3
- Random module
- Basic console input/output

---

4. Methodology

1. Create a list of five predefined words.
2. Randomly select one word as the secret word.
3. Show blanks (_) for each letter.
4. Allow the user to guess letters.
5. Update the blanks when guesses are correct.
6. Reduce attempts when guesses are wrong.
7. Display win/lose message.

---

5. Code Explanation

- The random.choice() function selects a word.
- A list of underscores (_) represents hidden letters.
- A while loop continues until the player wins or runs out of attempts.
- Letters are replaced in the word when correctly guessed.

---

6. Output

The program runs in the terminal/command prompt and displays:

- Hidden word
- Correct/incorrect messages
- Remaining attempts
- Win/Lose result

---

7. Conclusion

This project helped understand basic Python concepts such as loops, conditionals, strings, and lists. The Hangman game is a fun way to practice logical thinking and programming skills.

---

8. Future Enhancements

- Add difficulty levels
- Add ASCII art for the hangman
- Allow the user to add custom words
- Add GUI (Tkinter or Pygame)

PYTHON CODE –

```
import random


# List of 5 predefined words
words = ["apple", "banana", "python", "school", "happy"]


# Randomly choose a word
secret_word = random.choice(words)


# Create blanks for the secret word
guessed_word = ["_"] * len(secret_word)


# Track guessed letters and remaining attempts
attempts_left = 6
```

```
guessed_letters = []

print("Welcome to Hangman!")
print("Guess the word:", " ".join(guessed_word))

while attempts_left > 0:
    guess = input("Enter a letter: ").lower()

    # Check valid input
    if len(guess) != 1 or not guess.isalpha():
        print("Please enter a single alphabet letter.")
        continue

    if guess in guessed_letters:
        print("You already guessed this letter!")
        continue

    guessed_letters.append(guess)

    # If guess is correct
    if guess in secret_word:
        print("Good job! Letter found.")
        for i in range(len(secret_word)):
            if secret_word[i] == guess:
                guessed_word[i] = guess
    else:
        attempts_left -= 1
        print("Wrong guess! Attempts left:", attempts_left)

    print("Current word:", " ".join(guessed_word))

    # Check if player won
```

```
    if "_" not in guessed_word:
        print("🎉 Congratulations! You guessed the word:", secret_word)
        break

# If player runs out of attempts
if attempts_left == 0:
    print("Game Over! The word was:", secret_word)
```

VARIABLE DESCRIPTION

Variable Name	Data Type	Description / Purpose
words	List	Stores the predefined list of 5 words used in the game
secret_word	String	Stores the randomly selected word that the player needs to guess
guessed_word	List	A list containing _ (underscores) to represent unguessed letters of the secret word
attempts_left	Integer	Stores the number of remaining incorrect attempts (initially 6)
guessed_letters	List	Stores all letters guessed by the player (to avoid repetition)
guess	String	Stores the letter entered by the user
i	Integer (loop counter)	Used in loop to check each character of the secret word

OUTPUT SCREENSHOT