**TASK 1: Hangman Game**

**Goal:** Create a simple text-based Hangman game where the player guesses a word one letter at a time.

**Simplified Scope:**

* Use a small list of 5 predefined words (no need to use a file or API).
* Limit incorrect guesses to 6.
* Basic console input/output — no graphics or audio.

**Key Concepts Used:** **random**, **while** loop, **if-else**, **strings**, **lists**.

→

import random

def hangman():

# Predefined word list

words = ["apple", "banana", "grape", "orange", "mango"]

# Randomly choose a word

word = random.choice(words)

guessed\_word = ["\_"] \* len(word) # Underscores for hidden letters

guessed\_letters = [] # Track guessed letters

attempts = 6 # Limit of wrong guesses

print("🎮 Welcome to Hangman!")

print("Guess the word:")

print(" ".join(guessed\_word))

# Game loop

while attempts > 0 and "\_" in guessed\_word:

guess = input("Enter a letter: ").lower()

# Check for valid input

if len(guess) != 1 or not guess.isalpha():

print("⚠️ Please enter only a single letter.")

continue

if guess in guessed\_letters:

print("❗ You already guessed that letter.")

continue

guessed\_letters.append(guess)

if guess in word:

# Reveal the correct guessed letters

for i in range(len(word)):

if word[i] == guess:

guessed\_word[i] = guess

print("✅ Correct guess!")

else:

attempts -= 1

print(f"❌ Wrong guess! Attempts left: {attempts}")

print(" ".join(guessed\_word))

# Game over conditions

if "\_" not in guessed\_word:

print(f"🎉 Congratulations! You guessed the word: {word}")

else:

print(f"💀 Game Over! The word was: {word}")

# Run the game

hangman()

**TASK 2: Stock Portfolio Tracker**

* **Goal:** Build a simple stock tracker that calculates total investment based on manually defined stock prices.
* **Simplified Scope:**
  + User inputs stock names and quantity.
  + Use a hardcoded dictionary to define stock prices (e.g., {"AAPL": 180, "TSLA": 250}).
  + Display total investment value and optionally save the result in a .txt or .csv file.
* **Key Concepts Used:** **dictionary**, **input/output**, **basic arithmetic**, **file handling (optional)**.

→

def stock\_portfolio\_tracker():

# Hardcoded stock prices

stock\_prices = {

"AAPL": 180,

"TSLA": 250,

"GOOG": 2700,

"AMZN": 140,

"MSFT": 330

}

portfolio = {} # Dictionary to store user’s stock holdings

print("📈 Welcome to Stock Portfolio Tracker")

print("Available stocks and their prices:")

for stock, price in stock\_prices.items():

print(f"{stock}: ${price}")

while True:

stock = input("Enter stock symbol (or type 'done' to finish): ").upper()

if stock == "DONE":

break

if stock not in stock\_prices:

print("⚠️ Stock not available in list.")

continue

try:

qty = int(input(f"Enter quantity of {stock}: "))

portfolio[stock] = portfolio.get(stock, 0) + qty

except ValueError:

print("❗ Please enter a valid number.")

# Calculate total investment

total\_value = 0

print("\nYour Portfolio Summary:")

for stock, qty in portfolio.items():

value = stock\_prices[stock] \* qty

total\_value += value

print(f"{stock}: {qty} shares × ${stock\_prices[stock]} = ${value}")

print(f"\n💰 Total Investment Value: ${total\_value}")

# Optionally save to file

save\_choice = input("Do you want to save your portfolio to a file? (y/n): ").lower()

if save\_choice == "y":

with open("portfolio.txt", "w") as f:

f.write("Stock Portfolio Summary\n")

for stock, qty in portfolio.items():

f.write(f"{stock}: {qty} × ${stock\_prices[stock]} = ${stock\_prices[stock] \* qty}\n")

f.write(f"\nTotal Investment: ${total\_value}")

print("✅ Portfolio saved to portfolio.txt")

# Run the tracker

stock\_portfolio\_tracker()

### **TASK 3: Task Automation with Python Scripts**

**Goal:** Automate a small, real-life repetitive task.

**Pick One of These Ideas:**

* Move all **.jpg** files from a folder to a new folder.
* Extract all email addresses from a **.txt** file and save them to another file.
* Scrape the title of a fixed webpage and save it.

**Key Concepts Used:** **os**, **shutil**, **re**, **requests**, **file handling**.

→

import os

import shutil

def move\_jpg\_files(source\_folder, destination\_folder):

if not os.path.exists(destination\_folder):

os.makedirs(destination\_folder)

for file\_name in os.listdir(source\_folder):

if file\_name.lower().endswith(".jpg"):

source\_path = os.path.join(source\_folder, file\_name)

destination\_path = os.path.join(destination\_folder, file\_name)

shutil.move(source\_path, destination\_path)

print(f"✅ Moved: {file\_name}")

print("\n🎉 Task Completed! All .jpg files moved successfully.")

# Automatically get your user directory

home = os.path.expanduser("~")

source = os.path.join(home, "Downloads") # e.g., C:\Users\Prisha\Downloads

destination = os.path.join(home, "Pictures", "JPG\_Files")

move\_jpg\_files(source, destination)

* Query successful

### **TASK 4: Basic Chatbot**

**Goal:** Build a simple rule-based chatbot.

**Scope:**

* Input from user like: "**hello**", "**how are you**", "**bye**".
* Predefined replies like: "**Hi!**", "**I'm fine, thanks!**", "**Goodbye!**".

**Key Concepts Used:** **if-elif**, **functions**, **loops**, **input/output**.

→

def chatbot():

print("🤖 Chatbot: Hello! Type 'bye' to exit.")

while True:

user\_input = input("You: ").lower()

if user\_input == "hello":

print("🤖 Chatbot: Hi!")

elif user\_input == "how are you":

print("🤖 Chatbot: I'm fine, thanks!")

elif user\_input == "bye":

print("🤖 Chatbot: Goodbye! 👋")

break

else:

print("🤖 Chatbot: Sorry, I don't understand that.")

# Run the chatbot

chatbot()