***“ PHYTHON PROGRAMMING”***

***Tasks Project During Internship by Code Alpha***

**TASK : 1**

**Hangman Game Code:**

**import random**

**def hangman():**

**word\_list = ['python', 'java', 'kotlin', 'javascript', 'hangman']**

**word = random.choice(word\_list)**

**word\_completion = '\_' \* len(word)**

**guessed = False**

**guessed\_letters = []**

**guessed\_words = []**

**tries = 6**

**print("Let's play Hangman!")**

**print(display\_hangman(tries))**

**print(word\_completion)**

**print("\n")**

**while not guessed and tries > 0:**

**guess = input("Please guess a letter or word: ").lower()**

**if len(guess) == 1 and guess.isalpha():**

**if guess in guessed\_letters:**

**print(f"You already guessed the letter '{guess}'")**

**elif guess not in word:**

**print(f"'{guess}' is not in the word.")**

**tries -= 1**

**guessed\_letters.append(guess)**

**else:**

**print(f"Good job! '{guess}' is in the word.")**

**guessed\_letters.append(guess)**

**word\_as\_list = list(word\_completion)**

**indices = [i for i, letter in enumerate(word) if letter == guess]**

**for index in indices:**

**word\_as\_list[index] = guess**

**word\_completion = ''.join(word\_as\_list)**

**if '\_' not in word\_completion:**

**guessed = True**

**elif len(guess) == len(word) and guess.isalpha():**

**if guess in guessed\_words:**

**print(f"You already guessed the word '{guess}'")**

**elif guess != word:**

**print(f"'{guess}' is not the word.")**

**tries -= 1**

**guessed\_words.append(guess)**

**else:**

**guessed = True**

**word\_completion = word**

**else:**

**print("Invalid guess.")**

**print(display\_hangman(tries))**

**print(word\_completion)**

**print("\n")**

**if guessed:**

**print(f"Congrats! You guessed the word '{word}'!")**

**else:**

**print(f"Sorry, you ran out of tries. The word was '{word}'. Better luck next time!")**

**def display\_hangman(tries):**

**stages = [**

**"""**

**--------**

**| |**

**| O**

**| \\|/**

**| |**

**| / \\**

**-**

**""",**

**"""**

**--------**

**| |**

**| O**

**| \\|/**

**| |**

**| /**

**-**

**""",**

**"""**

**--------**

**| |**

**| O**

**| \\|/**

**| |**

**|**

**-**

**""",**

**"""**

**--------**

**| |**

**| O**

**| \\|**

**| |**

**|**

**-**

**""",**

**"""**

**--------**

**| |**

**| O**

**| |**

**| |**

**|**

**-**

**""",**

**"""**

**--------**

**| |**

**| O**

**|**

**|**

**|**

**-**

**""",**

**"""**

**--------**

**| |**

**|**

**|**

**|**

**|**

**-**

**"""**

**]**

**return stages[tries]**

**# Start the game**

**hangman()**

**Sample Output:**

**Let's play Hangman!**

**--------**

**| |**

**|**

**|**

**|**

**|**

**-**

**\_\_**

**Please guess a letter or word: a**

**'a' is not in the word.**

**--------**

**| |**

**| O**

**|**

**|**

**|**

**-**

**\_\_**

**Please guess a letter or word: p**

**Good job! 'p' is in the word.**

**--------**

**| |**

**| O**

**|**

**|**

**|**

**-**

**p\_\_\_**

**Please guess a letter or word: o**

**Good job! 'o' is in the word.**

**--------**

**| |**

**| O**

**|**

**|**

**|**

**-**

**p\_o\_\_\_**

**TASK:2**

**Stock Portfila Tracker**

**Requirements:**

**You need to install yfinance to fetch real-time stock prices:**

**pip install yfinance**

**Stock Portfolio Tracker Code:**

**import yfinance as yf**

**class Stock:**

**def \_init\_(self, ticker, shares):**

**self.ticker = ticker.upper()**

**self.shares = shares**

**def current\_price(self):**

**# Get the current price using yfinance**

**stock\_data = yf.Ticker(self.ticker)**

**price = stock\_data.history(period="1d")['Close'][0]**

**return price**

**def value(self):**

**# Calculate the current value of the stock in the portfolio**

**return self.shares \* self.current\_price()**

**class Portfolio:**

**def \_init\_(self):**

**self.stocks = {}**

**def add\_stock(self, ticker, shares):**

**if ticker in self.stocks:**

**self.stocks[ticker].shares += shares**

**else:**

**self.stocks[ticker] = Stock(ticker, shares)**

**print(f"Added {shares} shares of {ticker}.")**

**def remove\_stock(self, ticker, shares):**

**if ticker in self.stocks and self.stocks[ticker].shares >= shares:**

**self.stocks[ticker].shares -= shares**

**print(f"Removed {shares} shares of {ticker}.")**

**if self.stocks[ticker].shares == 0:**

**del self.stocks[ticker]**

**else:**

**print("Not enough shares to remove or stock not in portfolio.")**

**def portfolio\_value(self):**

**total\_value = 0**

**for stock in self.stocks.values():**

**total\_value += stock.value()**

**return total\_value**

**def display\_portfolio(self):**

**print("\nCurrent Portfolio:")**

**for ticker, stock in self.stocks.items():**

**print(f"{ticker}: {stock.shares} shares, Current Value: ${stock.value():.2f}")**

**print(f"Total Portfolio Value: ${self.portfolio\_value():.2f}\n")**

**# Usage Example**

**if \_name\_ == "\_main\_":**

**portfolio = Portfolio()**

**# Add stocks to portfolio**

**portfolio.add\_stock('AAPL', 10) # 10 shares of Apple**

**portfolio.add\_stock('GOOGL', 5) # 5 shares of Google**

**portfolio.add\_stock('TSLA', 2) # 2 shares of Tesla**

**# Display portfolio**

**portfolio.display\_portfolio()**

**# Remove stock**

**portfolio.remove\_stock('AAPL', 5)**

**# Display updated portfolio**

**portfolio.display\_portfolio()**

**Output:**

**Added 10 shares of AAPL.**

**Added 5 shares of GOOGL.**

**Added 2 shares of TSLA.**

**Current Portfolio:**

**AAPL: 10 shares, Current Value: $1893.60**

**GOOGL: 5 shares, Current Value: $7140.75**

**TSLA: 2 shares, Current Value: $536.50**

**Total Portfolio Value: $9570.85**

**Removed 5 shares of AAPL.**

**Current Portfolio:**

**AAPL: 5 shares, Current Value: $946.80**

**GOOGL: 5 shares, Current Value: $7140.75**

**TSLA: 2 shares, Current Value: $536.50**

**Total Portfolio Value: $8623.95**

**Explanation of How the Tracker Works:**

**1. Stock Class:**

**The Stock class is responsible for fetching the current stock price using the yfinance library.**

**It also calculates the total value of the stock based on the number of shares owned.**

**2. Portfolio Class:**

**The Portfolio class stores a collection of stocks.**

**It allows you to add and remove stocks, and displays the total portfolio value based on real-time prices.**

**3. Portfolio Operations:**

**You can add stocks to the portfolio by specifying the stock ticker and number of shares.**

**You can also remove stocks, either fully or partially, from the portfolio.**

**The total value of the portfolio is calculated and displayed after each operation.**

**API Usage Notes:**

**yfinance: The prices fetched by yfinance are usually the latest closing prices, which might not reflect real-time prices if the market is closed.**

**Customization: You can enhance this program by tracking the purchase price, gains/losses, dividends, and other stock-related data**

**TASK:3**

**Basic Chatbox:**

**def chat():**

**print("Chatbot: Hi there! I'm your friendly chatbot. How can I help you today?")**

**while True:**

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

**if "hello" in user\_input or "hi" in user\_input:**

**print("Chatbot: Hello! How can I assist you?")**

**elif "how are you" in user\_input:**

**print("Chatbot: I'm just a program, but I'm functioning as expected! How about you?")**

**elif "bye" in user\_input or "exit" in user\_input:**

**print("Chatbot: Goodbye! Have a great day!")**

**break**

**elif "your name" in user\_input:**

**print("Chatbot: I'm a basic chatbot created to chat with you.")**

**elif "help" in user\_input:**

**print("Chatbot: I'm here to chat with you. Try saying 'hello', ask me 'how are you', or just say 'bye' to exit.")**

**else:**

**print("Chatbot: Sorry, I didn't understand that. Can you please rephrase?")**

**# Start the chat**

**if \_name\_ == "\_main\_":**

**chat()**

**Output:**

**Chatbot: Hi there! I'm your friendly chatbot. How can I help you today?**

**You: Hello**

**Chatbot: Hello! How can I assist you?**

**You: What's your name?**

**Chatbot: I'm a basic chatbot created to chat with you.**

**You: How are you?**

**Chatbot: I'm just a program, but I'm functioning as expected! How about you?**

**You: Help**

**Chatbot: I'm here to chat with you. Try saying 'hello', ask me 'how are you', or just say 'bye' to exit.**

**You: Bye**

**Chatbot: Goodbye! Have a great day!**

**How it works:**

**1. Greeting: The chatbot starts by greeting the user and asking how it can help.**

**2. User Input: The chatbot continuously waits for user input.**

**3. Predefined Responses: It matches specific keywords from the user input (like "hello", "how are you", "bye", etc.) and responds accordingly.**

**4. Exit Condition: If the user types "bye" or "exit", the chatbot will say goodbye and end the conversation.**

**Features:**

**Simple Responses: The chatbot can respond to a few basic phrases.**

**Loop: It keeps asking for input until the user types "bye" or "exit".**

***THANKYOU***