LAB ASSIGNMNET 9.5

Subject : AI Assistant Coding

Hall Ticket No : 2303A52465

Name : M.Avinash

Batch : 50

Date : 20/2/26

Task-1 :

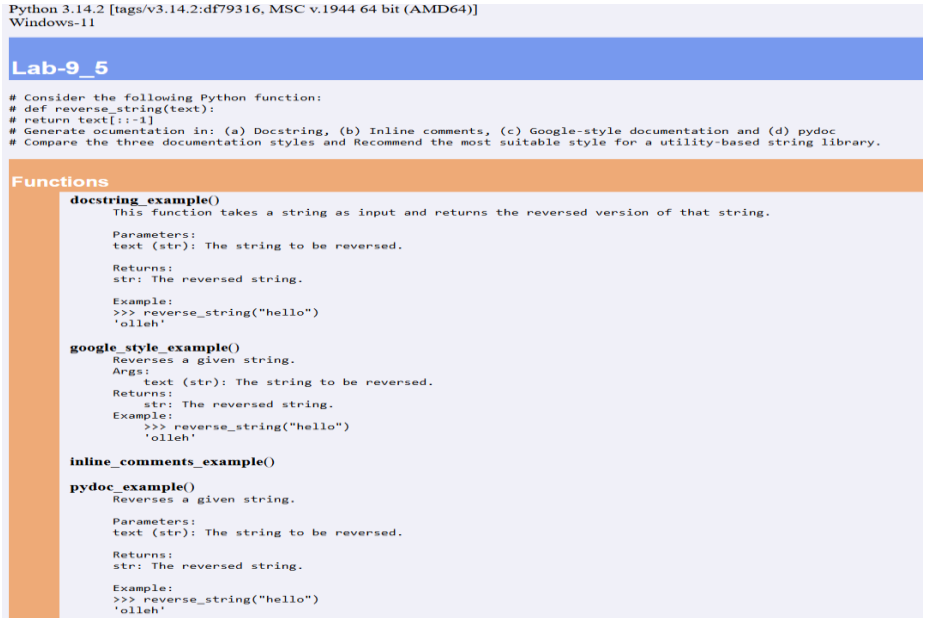
Prompt : # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for reversing a string function

```python
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for reversing a string function
def reverse_string(s):
    """
    Reverses the given string.

    Args:
        s (str): The string to be reversed.

    Returns:
        str: The reversed string.
    """
    # Initialize an empty string to store the reversed result
    reversed_str = ""

    # Iterate through the input string in reverse order
    for char in s[::-1]:
        # Append each character to the reversed string
        reversed_str += char

    return reversed_str
# Example usage
if __name__ == "__main__":
    input_string = "Hello, World!"
    print(f"Original string: {input_string}")
    print(f"Reversed string: {reverse_string(input_string)}")
```

Output :

```
NAME
    Lab-9_5

DESCRIPTION
    # Consider the following Python function:
    # def reverse_string(text):
    # return text[::-1]
    # Generate ocumentation in: (a) Docstring, (b) Inline comments, (c) Google-style documentation and (d) pyd
    # Compare the three documentation styles and Recommend the most suitable style for a utility-based string

FUNCTIONS
    docstring_example()
        This function takes a string as input and returns the reversed version of that string.

        Parameters:
        text (str): The string to be reversed.

        Returns:
        str: The reversed string.

        Example:
        >>> reverse_string("hello")
        'olleh'

    google_style_example()
        Reverses a given string.
        Args:
            text (str): The string to be reversed.
        Returns:
            str: The reversed string.
        Example:
            >>> reverse_string("hello")
            'olleh'

    inline_comments_example()

    pydoc_example()
        Reverses a given string.

        Parameters:
        text (str): The string to be reversed.
```

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10>
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10>
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -w Lab-9_5
wrote Lab-9_5.html
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -p 1234
Server ready at http://localhost:1234/
Server commands: [b]rowser, [q]uit
server> b
server>
```



Python 3.14.2 [tags/v3.14.2:df79316, MSC v.1944 64 bit (AMD64)]
Windows-11

Lab-9_5

```
# Consider the following Python function:
# def reverse_string(text):
# return text[::-1]
# Generate ocumentation in: (a) Docstring, (b) Inline comments, (c) Google-style documentation and (d) pydoc
# Compare the three documentation styles and Recommend the most suitable style for a utility-based string library.
```

Functions

docstring_example()

```
This function takes a string as input and returns the reversed version of that string.

Parameters:
text (str): The string to be reversed.

Returns:
str: The reversed string.

Example:
>>> reverse_string("hello")
'olleh'
```

google_style_example()

```
Reverses a given string.
Args:
    text (str): The string to be reversed.
Returns:
    str: The reversed string.
Example:
    >>> reverse_string("hello")
    'olleh'
```

inline_comments_example()

pydoc_example()

```
Reverses a given string.

Parameters:
text (str): The string to be reversed.

Returns:
str: The reversed string.

Example:
>>> reverse_string("hello")
'olleh'
```

Task-2 :

Prompt : Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Password Strength Checker

```python
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Password Strength Check
def check_password_strength(password):
    """
    Checks the strength of a given password.

    Args:
        password (str): The password to be evaluated.

    Returns:
        str: A message indicating the strength of the password.
    """
    # Initialize strength score
    strength_score = 0

    # Check for length of the password
    if len(password) >= 8:
        strength_score += 1
    if len(password) >= 12:
        strength_score += 1

    # Check for uppercase letters
    if any(char.isupper() for char in password):
        strength_score += 1

    # Check for lowercase letters
    if any(char.islower() for char in password):
        strength_score += 1
```

Output :

```
NAME
    Lab-9 - # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Password Strength Checke

FUNCTIONS
    check_password_strength(password)
        Checks the strength of a given password.

        Args:
            password (str): The password to be evaluated.

        Returns:
            str: A message indicating the strength of the password.

FILE
    c:\users\gopal\onedrive\desktop\ai ac\lab-10\lab-9.py
```

Lab-9

```
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Password Strength Checker
```

Functions

check_password_strength(password)

```
Checks the strength of a given password.

Args:
    password (str): The password to be evaluated.

Returns:
    str: A message indicating the strength of the password.
```

Task-3 :

Prompt : Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Math Utilities Module

```python
# # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Math Utilities Module
def add(a, b):
    """
    Adds two numbers.

    Args:
        a (float): The first number.
        b (float): The second number.

    Returns:
        float: The sum of the two numbers.
    """
    return a + b

def subtract(a, b):
    """
```

Output :

```
NAME
    Lab-9 - # # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Math Utilities Module

FUNCTIONS
    add(a, b)
        Adds two numbers.

        Args:
            a (float): The first number.
            b (float): The second number.
```

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -w Lab-9
wrote Lab-9.html
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -p 2412
Server ready at http://localhost:2412/
Server commands: [b]rowser, [q]uit
server> b
```

Lab-9

Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Math Utilities Module

Functions

add(a, b)
Adds two numbers.

Args:
a (float): The first number.
b (float): The second number.

Returns:
float: The sum of the two numbers.

divide(a, b)
Divides the first number by the second number.

Args:
a (float): The numerator.
b (float): The denominator.

Task-4 :

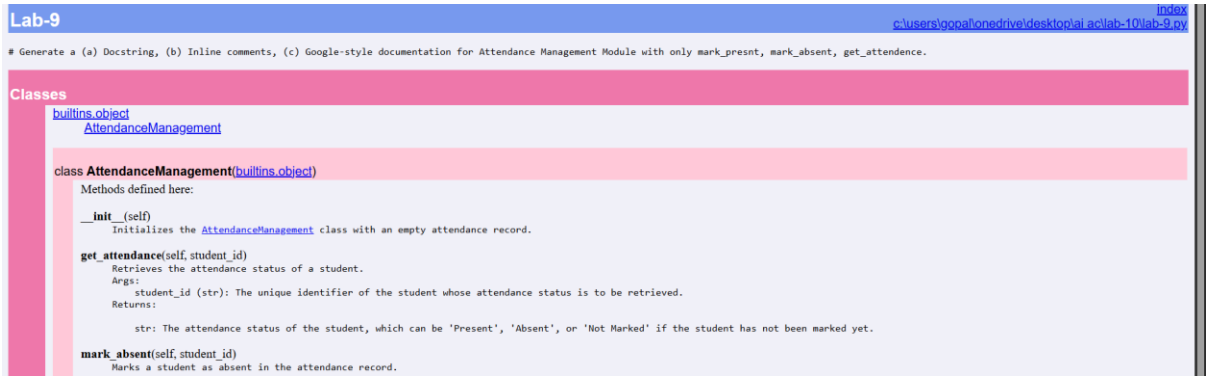
Prompt : Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Attendance Management Module with only mark_presnt, mark_absent, get_attendence.

```python
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Attendance Management Module with only mark_presnt, mark_absent, get_attendence.

class AttendanceManagement:
    def __init__(self):
        """
        Initializes the AttendanceManagement class with an empty attendance record.
        """
        self.attendance_record = {}
    def mark_present(self, student_id):
        """
        Marks a student as present in the attendance record.

        Args:
            student_id (str): The unique identifier of the student to be marked as present.
        """
        self.attendance_record[student_id] = 'Present'
    def mark_absent(self, student_id):
        """
        Marks a student as absent in the attendance record.

        Args:
            student_id (str): The unique identifier of the student to be marked as absent.
        """
        self.attendance_record[student_id] = 'Absent'
    def get_attendance(self, student_id):
        """
        Retrieves the attendance status of a student.
        Args:
            student_id (str): The unique identifier of the student whose attendance status is to be retrieved.
        Returns:

            str: The attendance status of the student, which can be 'Present', 'Absent', or 'Not Marked' if the student has not been marked yet.
        """
```

Output :

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc Lab-9

NAME
    Lab-9 - # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Attendance Management Module with only mark_presnt, mark_absent, get_attendence.

CLASSES
    builtins.object
        AttendanceManagement

    class AttendanceManagement(builtins.object)
     |  Methods defined here:
     |
```

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -w Lab-9
Present
wrote Lab-9.html
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -p 2419
Server ready at http://localhost:2419/
Server commands: [b]rowser, [q]uit
server> b
```



Lab-9

index
c:\users\gopal\onedrive\desktop\ai ac\lab-10\lab-9.py

Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for Attendance Management Module with only mark_presnt, mark_absent, get_attendence.

Classes

builtins.object
AttendanceManagement

class **AttendanceManagement**(builtins.object)

Methods defined here:

__init__(self)
Initializes the AttendanceManagement class with an empty attendance record.

get_attendance(self, student_id)
Retrieves the attendance status of a student.
Args:
student_id (str): The unique identifier of the student whose attendance status is to be retrieved.
Returns:

str: The attendance status of the student, which can be 'Present', 'Absent', or 'Not Marked' if the student has not been marked yet.

mark_absent(self, student_id)
Marks a student as absent in the attendance record.

Task-5 :

Prompt : Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for File Handling Function.

```python
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for File Handling Function.
def read_file(file_path):
    """
    Reads the content of a file and returns it as a string.

    Args:
        file_path (str): The path to the file to be read.
    Returns:
        str: The content of the file as a string.
    Raises:
        FileNotFoundError: If the specified file does not exist.
        IOError: If there is an error reading the file.
    """
    try:
        with open(file_path, 'r') as file:
            content = file.read()  # Read the entire content of the file
            return content  # Return the content as a string
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
    except IOError as e:
        print(f"Error reading the file '{file_path}': {e}")
# Example usage:
# file_content = read_file('example.txt')
```

Output :

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc Lab-9
Help on module Lab-9:

NAME
    Lab-9 - # Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for File Handling Function.

FUNCTIONS
    read_file(file_path)
        Reads the content of a file and returns it as a string.

        Args:
            file_path (str): The path to the file to be read.
        Returns:
            str: The content of the file as a string.
        Raises:
            FileNotFoundError: If the specified file does not exist.
            IOError: If there is an error reading the file.

FILE
    c:\users\gopal\onedrive\desktop\ai ac\lab-10\lab-9.py
```

```
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -w Lab-9
wrote Lab-9.html
PS C:\Users\GOPAL\OneDrive\Desktop\AI AC\Lab-10> python -m pydoc -p 2427
Server ready at http://localhost:2427/
Server commands: [b]rowser, [q]uit
server> b
```

Lab-9

```
# Generate a (a) Docstring, (b) Inline comments, (c) Google-style documentation for File Handling Function.
```

Functions

read_file(file_path)

```
Reads the content of a file and returns it as a string.

Args:
    file_path (str): The path to the file to be read.
Returns:
    str: The content of the file as a string.
Raises:
    FileNotFoundError: If the specified file does not exist.
    IOError: If there is an error reading the file.
```