TECHNIK NEST
INNOVATIVE MINDS, NESTING SUCCESS

Name: Musfira Ahmed

Intern ID: TN/IN01/PY/002

Email ID : ahmedmusfira3@gmail.com

Internship Domain : Python Development

Task Week : 04

Instructor Name : Mr Hassan Ali

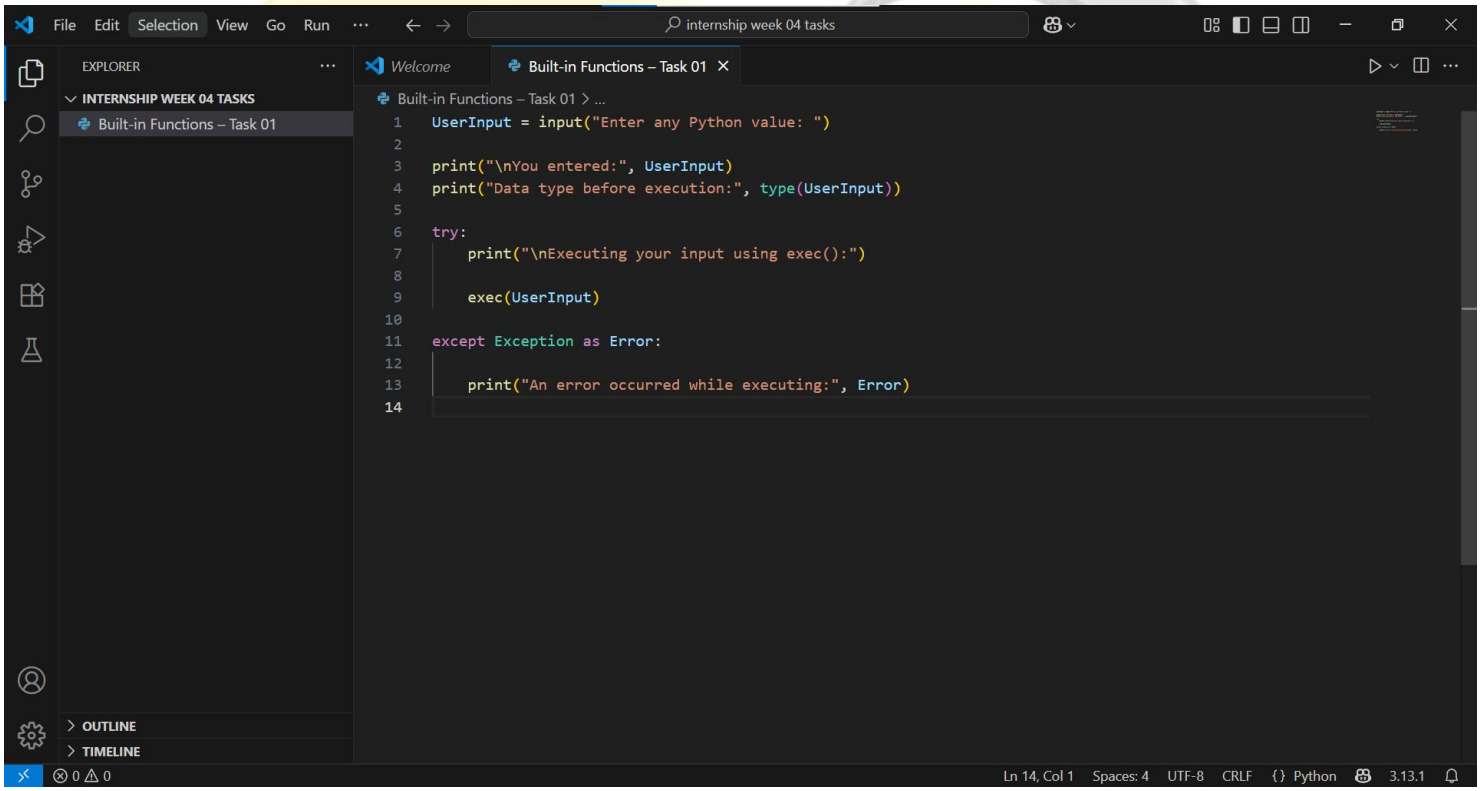
Task 1 :

Ask user to input any value. Use type() to check its data type. Use exec() to execute a string as Python code.

What I Did (Step by Step):

- Took input from the user using input().
- Displayed the input and its type using type().
- Used exec() to execute the input as Python code.
- Added try-except to handle execution errors.
- Printed any output or error from the execution.

Code Screenshots



Output Screenshot

```
PROBLEMS   OUTPUT   TERMINAL   PORTS
TERMINAL                                                    Python
ask 01"
Enter any Python value: print ("Hello")

You entered: print ("Hello")
Data type before execution: <class 'str'>

Executing your input using exec():
Hello
PS D:\internship week 04 tasks>
Ln 14, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges:

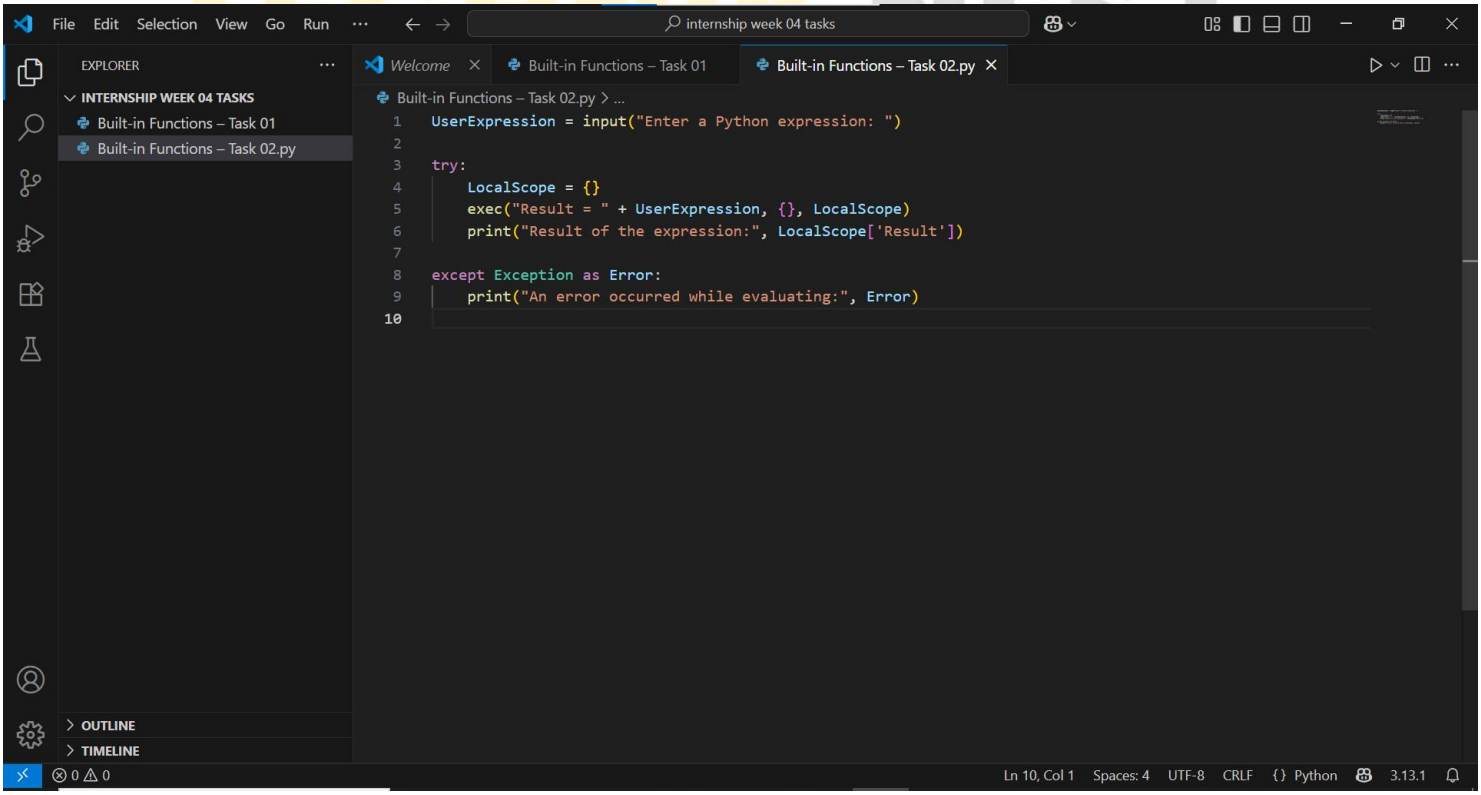
- Learned that all input() values are strings by default.
- Understood how exec() runs dynamic Python code.
- Faced issues when invalid code caused syntax errors.
- Used error handling to avoid crashes during execution.
- Realized exec() can be dangerous if not used safely.

Task 02:

What I Did (Step by Step):

- Took a math expression as a string input from the user.
- Combined it into a statement using exec() to assign the result to a variable.
- Executed it safely using a try-except block.
- Retrieved the result from the dynamically created variable.
- Displayed the result or any execution error.

Code Screenshots



Output Screenshot

```
PROBLEMS   OUTPUT   TERMINAL   PORTS
TERMINAL                                                    Python
PS D:\internship week 04 tasks> & "C:/Program Files/Python313/python.exe" "d:/internship week 04 tasks/Built-in Functions - T
ask 02.py"
Enter a Python expression: 2+5*4
Result of the expression: 22
PS D:\internship week 04 tasks>
Ln 10, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges:

- Learned how to evaluate expressions using exec() by building an assignment string.
- Understood that exec() doesn't return values directly like eval().
- Faced a challenge accessing variables created inside exec().
- Solved it by assigning to a known variable name like Result.
- Realized that eval() is simpler for expressions, but exec() offers more flexibility.

<u>Task 03:</u>

Use math module to take a radius input from user and calculate:
- Area of circle, circumference, and square root of area.

What I Did (Step by Step):

- Imported the math module for advanced mathematical functions.
- Took radius input from the user and converted it to float.
- Calculated area using πr^2 and circumference using $2\pi r$.
- Found square root of area using math.sqrt().
- Printed all three calculated values clearly.

Code Screenshots

```python

Radius = float(input("Enter the radius of the circle: ")) # take input from user

Area = math.pi * Radius ** 2
Circumference = 2 * math.pi * Radius
SqrtArea = math.sqrt(Area)

# print the results

print(f"Area of Circle: {Area:.5f}")
print(f"Circumference: {Circumference:.5f}")
print(f"Square Root of Area: {SqrtArea:.5f}")
```

Output Screenshot

```
PS D:\internship week 04 tasks> & "C:/Program Files/Python313/python.exe" "d:/internship week 04 tasks/math and random Modules - Task 01.py"
Enter the radius of the circle: 45
Area of Circle: 6361.72512
Circumference: 282.74334
Square Root of Area: 79.76042
PS D:\internship week 04 tasks>
```

Learnings and Challenges:

- Learned how to use math.pi for accurate π value.
- Understood the importance of data type conversion (float).
- Faced issue with square root before calculating area first.
- Practiced chaining formulas and printing multiple outputs.
- Strengthened understanding of geometry in real Python usage.

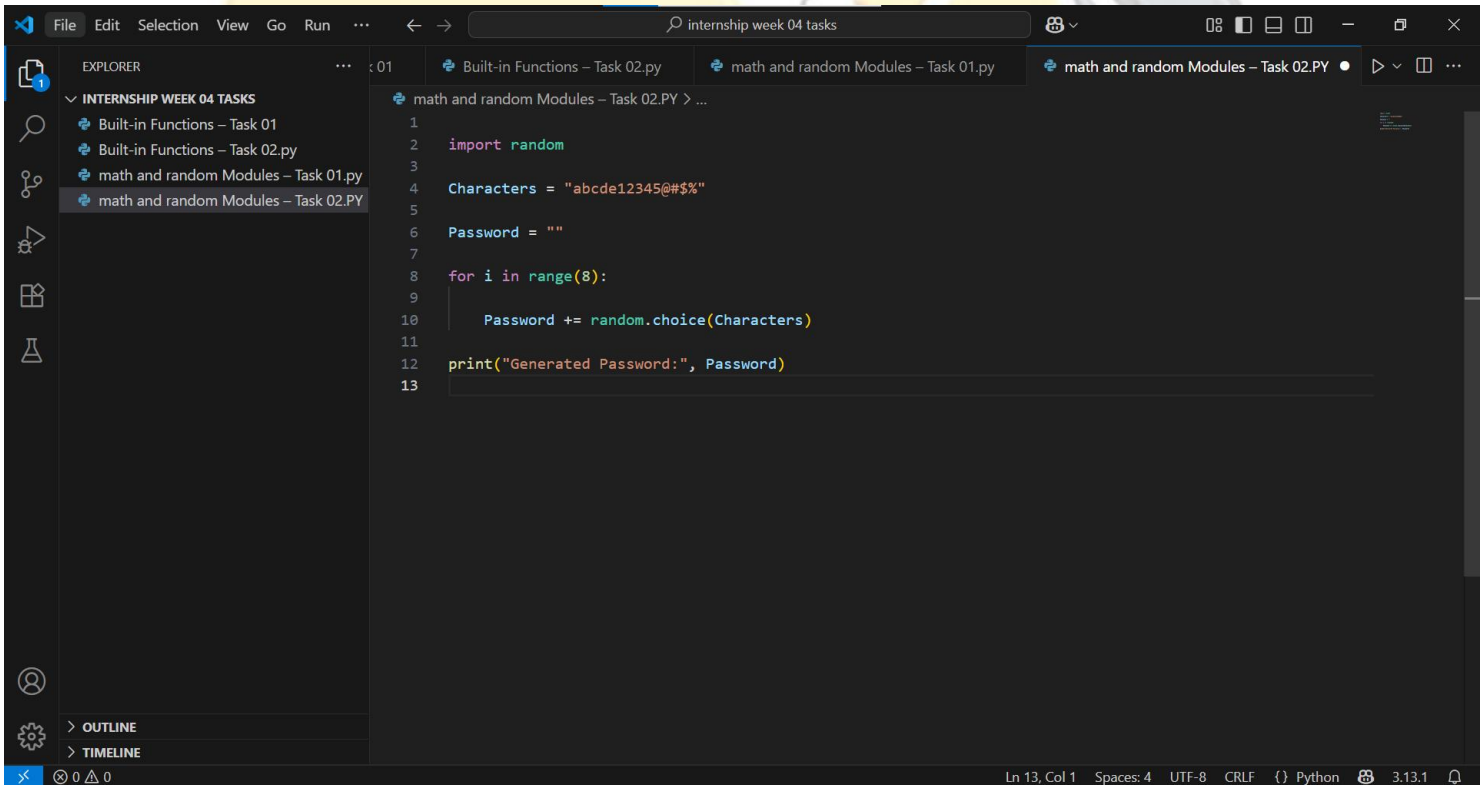
Task 04:

Use random module to generate a random 8-character password using letters, numbers, and symbols.

What I Did (Step by Step):

- Used random module to pick random characters.
- Created a simple string with letters, numbers, and symbols.
- Initialized an empty password variable.
- Used a loop to add 8 random characters.
- Printed the final password to the screen.

Code Screenshots



Output Screenshots

```
PROBLEMS    OUTPUT    TERMINAL    PORTS
TERMINAL
PS D:\internship week 04 tasks> & "C:/Program Files/Python313/python.exe" "d:/internship week 04 tasks/math and random Modu
les - Task 02.PY"
Generated Password: c1ca2@aa
PS D:\internship week 04 tasks>
Ln 5, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges

- Learned how to use random.choice() in a loop.
- Understood string concatenation to build passwords.
- Practiced simple loop logic.
- Faced small confusion with string vs list (but fixed it).
- Gained confidence using print() and variables.

<u>Task 05:</u>

Using datetime module, ask user for their birth date and show:
- Their age in years and number of days lived.

What I Did (Step by Step):

- Imported the datetime module to use dates.
- Asked the user for their birth year, month, and day.
- Created their birthdate and got today's date.
- Calculated their age in years and total days lived.
- Printed the age and number of days to the screen.

Code Screenshots

```python
import datetime

Year = int(input("Enter your birth year:"))
Month = int(input("Enter your birth month:"))
Day = int(input("Enter your birth day:"))

BirthDate = datetime.date(Year, Month, Day)
Today = datetime.date.today()

AgeYears = Today.year - BirthDate.year

if (Today.month, Today.day) < (BirthDate.month, BirthDate.day):
    AgeYears -= 1

DaysLived = (Today - BirthDate).days

print("You are", AgeYears, "years old.")
print("You have lived for", DaysLived, "days.")
```

Output Screenshots

```
PS D:\internship week 04 tasks> & "C:/Program Files/Python313/python.exe" "d:/internship week 04 tasks/datetime, os, re Modules - Tasks 01.py"
Enter your birth year:2005
Enter your birth month:3
Enter your birth day:14

You are 20 years old.
You have lived for 7425 days.
PS D:\internship week 04 tasks>
```

Learning and Challenges

- Learned how to work with real dates using datetime.
- Practiced asking for multiple inputs from the user.
- Faced a small challenge checking if birthday passed this year.
- Learned how to subtract dates to get days difference.
- Understood how age calculation can change based on the date.

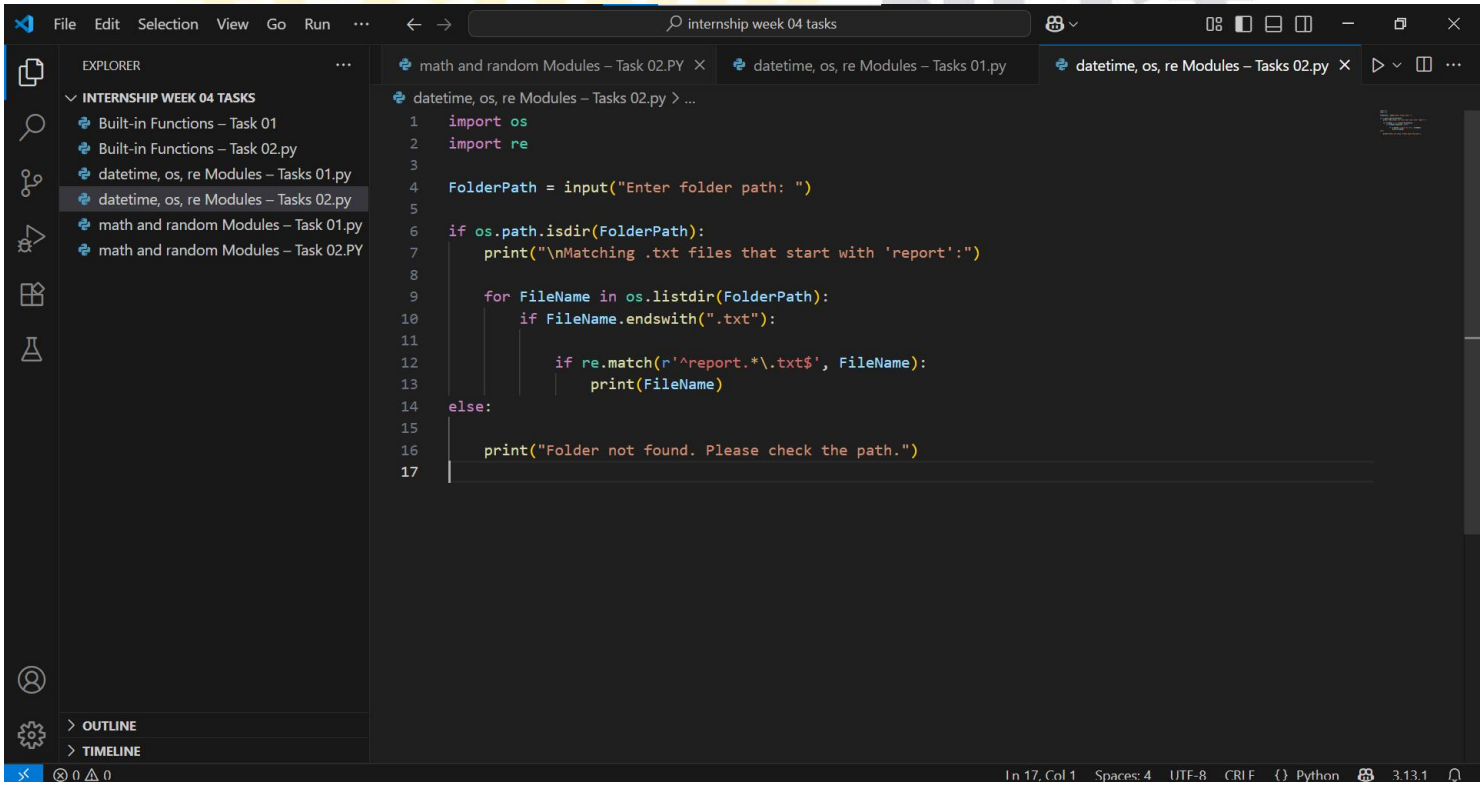
Task 06:

Create a script using os and re that lists all `.txt` files from a folder and filters only those that match a pattern (e.g., start with 'report').

What I Did (Step by Step):

- Imported os to list files in a folder.
- Used re (regular expressions) to match filenames.
- Asked the user to enter a folder path.
- Checked if file ends with .txt and starts with 'report'.
- Displayed the matched .txt files.

Code Screenshots



Output Screenshots

```
PROBLEMS    OUTPUT    TERMINAL    PORTS
TERMINAL                                                    Python
PS D:\internship week 04 tasks> & "C:/Program Files/Python313/python.exe" "d:/internship week 04 tasks/datetime, os, re Mod
ules - Tasks 02.py"
Enter folder path: D:\TestFiles

Matching .txt files that start with 'report':
report1.txt.txt
reportdata.txt.txt
report_final.txt.txt
PS D:\internship week 04 tasks>
Ln 17, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges

- Learned how to list files using os.listdir().
- Practiced using regex to filter by pattern.
- Faced issues with case sensitivity in re.match() (can fix with flags).
- Understood how to combine conditions (endswith + regex).
- Gained confidence working with file and folder paths.