

Name : SAMRA NAZ

Intern ID : TN/IN01/PY/008

Email ID : naazsamra01@gmail.com

Internship Domain : python

Week 6 task

Instructor Name : Hassan ALI

- **Task 1:**

Use math & statistics libraries to get square roots and average.

Description:

This Python script uses the built-in `math` and `statistics` libraries to:

- Calculate the square roots of a list of numbers
- Find the average (mean) of those numbers

It prints the original numbers, their square roots, and the average to the screen.

Output

```
task1.py > ...
1  #Task 1:Use math & statistics libraries to get square roots and average.
2  import math          # for square root
3  import statistics    # for average
4
5  # List of numbers
6  numbers = [4, 9, 16, 25, 36]
7
8  # Get square roots
9  square_roots = [math.sqrt(num) for num in numbers]
10
11 # Calculate average
12 average = statistics.mean(numbers)
13
14 # Print results
15 print("Original numbers:", numbers)
16 print("Square roots:", square_roots)
17 print("Average:", average)
18
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\naaazs\Documents\week6task internship2> & C:/Python/Python313/python.exe "c:/t
Original numbers: [4, 9, 16, 25, 36]
Square roots: [2.0, 3.0, 4.0, 5.0, 6.0]
Average: 18
○ PS C:\Users\naaazs\Documents\week6task internship2>
```

- **Task 2:**

Create a custom package and import it in another script.

Description:

This task shows how to create a custom Python package named `mypackage` with a module `tools.py` that contains a `multiply()` function.

We then import this package in another script (`main.py`, placed outside the package folder) and use the function to perform multiplication and print the result.

Output

```
main.py > ...
1
2  ##Task 2:Create a custom package and import it in and
3  from mypackage import tools
4
5  result = tools.multiply(5, 7)
6
7  print("The result is:", result)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\naazs\Documents\week6task internship2> & C:/Python/Py
The result is: 35
○ PS C:\Users\naazs\Documents\week6task internship2> 
```

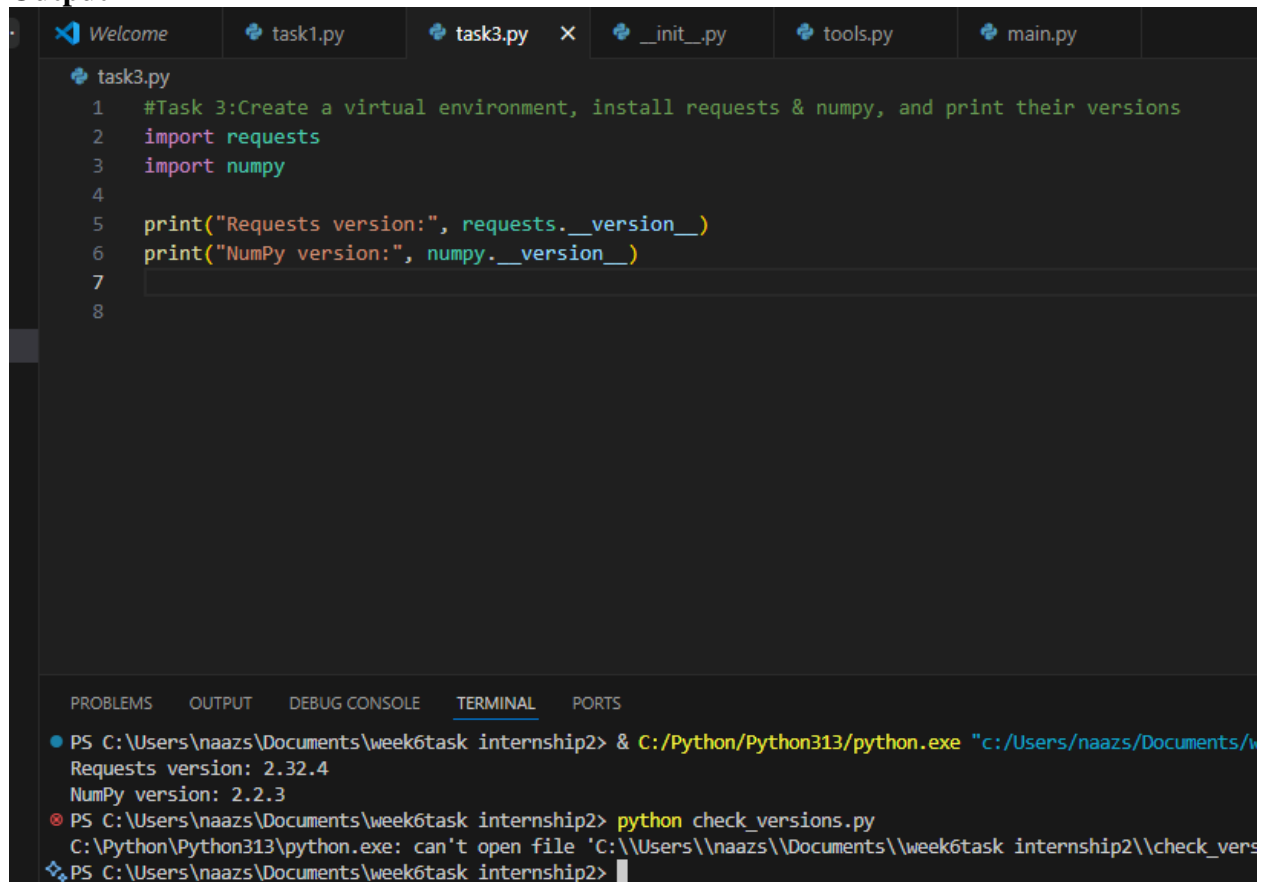
- **Task 3:**

Create a virtual environment, install requests & numpy, and print their versions.

description:

This task demonstrates how to set up a Python virtual environment, install external libraries (requests and numpy), and check their installed versions using a simple Python script.

Output



The screenshot shows a VS Code editor with a file explorer at the top containing 'Welcome', 'task1.py', 'task3.py', '.__init__.py', 'tools.py', and 'main.py'. The 'task3.py' file is open, showing the following code:

```
1 #Task 3:Create a virtual environment, install requests & numpy, and print their versions
2 import requests
3 import numpy
4
5 print("Requests version:", requests.__version__)
6 print("NumPy version:", numpy.__version__)
7
8
```

Below the editor is a terminal window with the following output:

```
PS C:\Users\naazs\Documents\week6task internship2> & C:/Python/Python313/python.exe "c:/Users/naazs/Documents/w
Requests version: 2.32.4
NumPy version: 2.2.3
PS C:\Users\naazs\Documents\week6task internship2> python check_versions.py
C:\Python\Python313\python.exe: can't open file 'C:\\Users\\naazs\\Documents\\week6task internship2\\check_vers
PS C:\Users\naazs\Documents\week6task internship2>
```

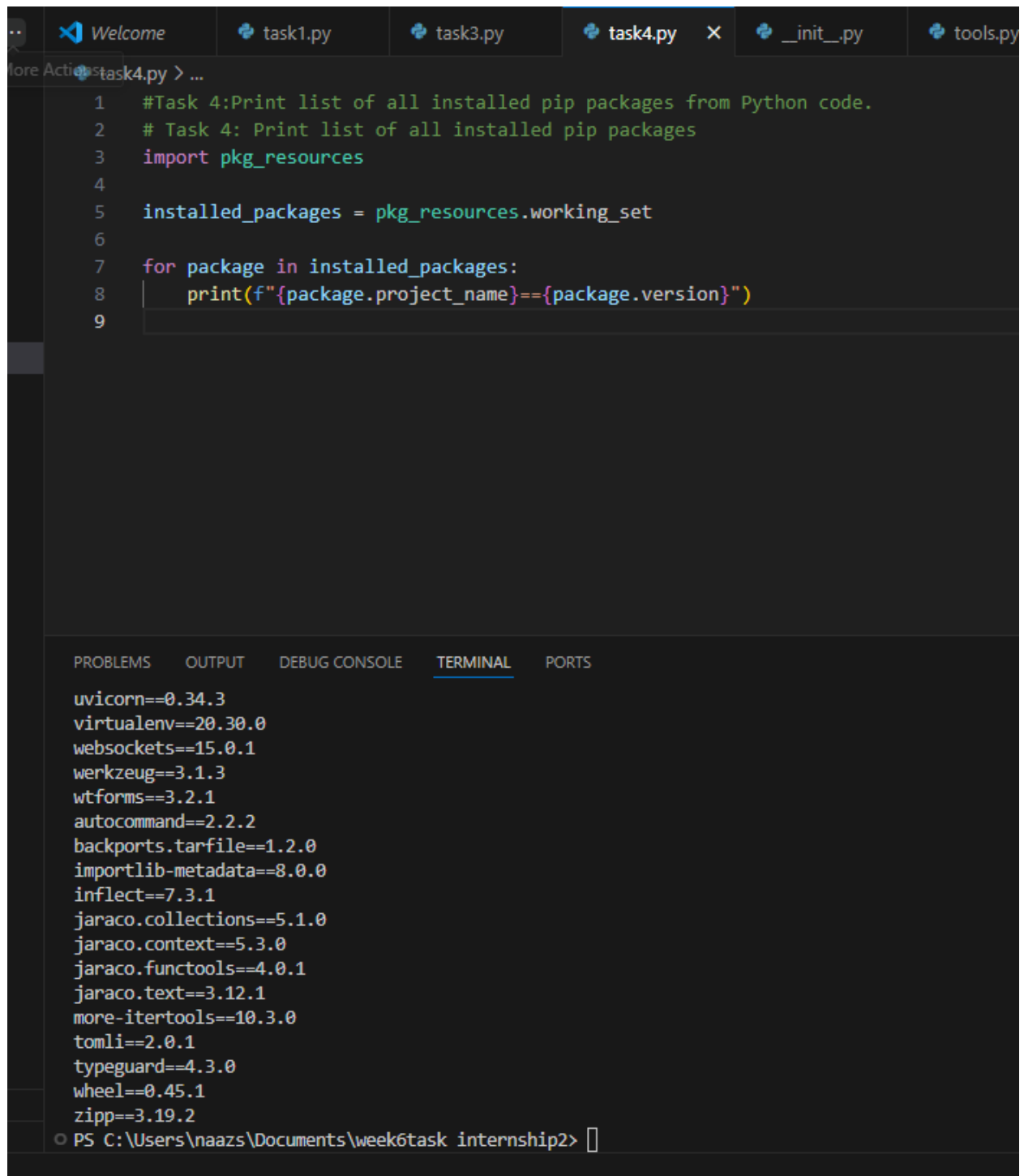
- **Task 4:**

Print list of all installed pip packages from Python code.

description:

This script uses the `pkg_resources` module from `setuptools` to list all installed pip packages and their versions in the current Python environment.

Output



The image shows a Visual Studio Code editor window with several tabs: 'Welcome', 'task1.py', 'task3.py', 'task4.py' (active), '__init__.py', and 'tools.py'. The active tab 'task4.py' contains the following Python code:

```
1 #Task 4:Print list of all installed pip packages from Python code.
2 # Task 4: Print list of all installed pip packages
3 import pkg_resources
4
5 installed_packages = pkg_resources.working_set
6
7 for package in installed_packages:
8     print(f"{package.project_name}=={package.version}")
9
```

Below the code editor, the 'TERMINAL' panel is open, displaying the output of the script. The output lists various installed packages and their versions, such as 'uvicorn==0.34.3', 'virtualenv==20.30.0', 'websockets==15.0.1', etc. The terminal prompt at the bottom indicates the current directory is 'C:\Users\naaazs\Documents\week6task internship2'.

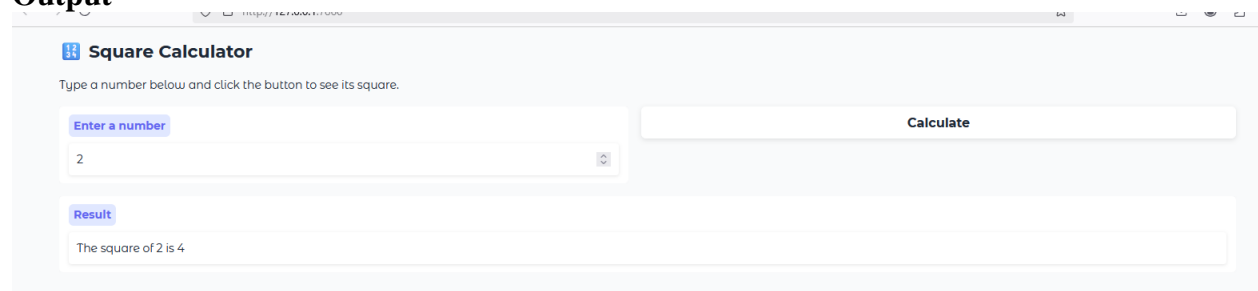
- **Task 5:**
Create Gradio app that takes a number and returns its square.

description:

This is a simple Gradio web app that asks the user to enter a number, then calculates and displays its square when the “**Calculate**” button is clicked.

- ✓ Built with Gradio’s `Blocks` layout for a clean and modern interface.
- ✓ Includes a title and instructions to guide the user.
- ✓ Shows the result neatly in a textbox below.
- ✓ Uses a soft theme to make the interface look smooth and user-friendly.

Output

The screenshot shows a web browser window displaying a Gradio application titled "Square Calculator". The interface has a light blue header with the title and a small icon. Below the header, there is a instruction: "Type a number below and click the button to see its square." The main area contains two input fields. The first is labeled "Enter a number:" and contains the value "2". To its right is a button labeled "Calculate". Below these is a second input field labeled "Result:" which displays the text "The square of 2 is 4". The interface is clean and modern, using a soft theme.

- **Task 6:**
Create Gradio interface that takes a sentence and returns it reversed.

description:

This Gradio app takes a sentence typed by the user and returns it reversed when the “**Reverse**” button is clicked.

- ✓ Built using Gradio’s `Blocks` layout for a clean and modern look.
- ✓ Includes a textbox for input, a button to trigger the action, and an output box to display the reversed sentence.
- ✓ Uses a soft theme to keep the interface simple and user-friendly.

output

Reverse Sentence App

Type any sentence below and see it reversed instantly!

Your Sentence

my name is samra naz

Reverse

Reversed Sentence

zan arnas si eman ym