

PRACTICAL FILE

OF

PYTHON PROGRAMMING

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Submitted by:

Name: LAKSH CHIKARA
(Roll No): 24/SCA/BCA(AI-ML)/035

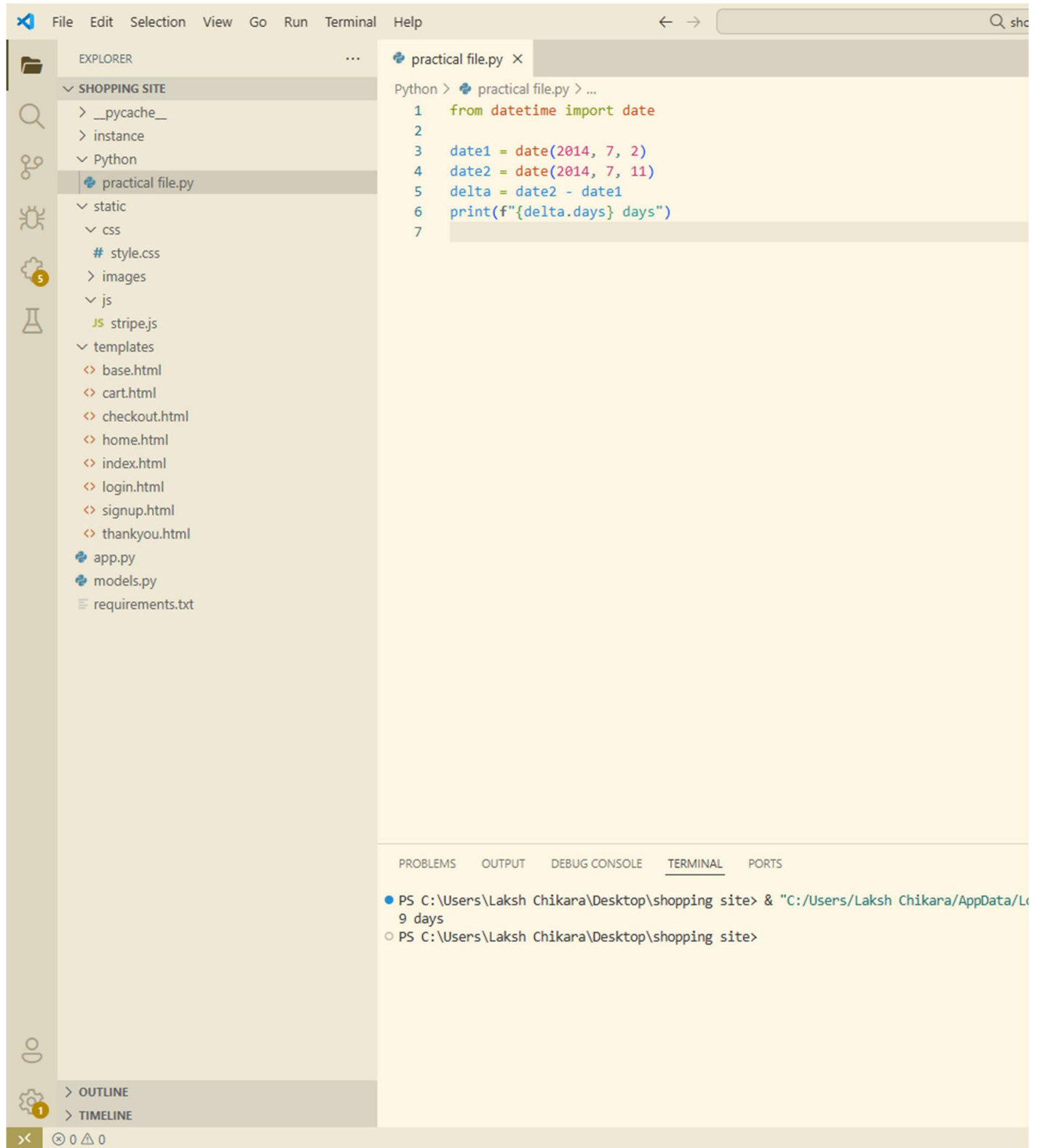


School of Computer Applications
Manav Rachna International Institute of Research and Studies
(DEEMED TO BE UNIVERSITY)

Sector-43, Aravalli Hills
Faridabad – 121001

October 2025

Ques: 1) Write a Python program to calculate number of days between two dates. Sample dates: (2014, 7, 2), (2014, 7,11) Expected output : 9 days.



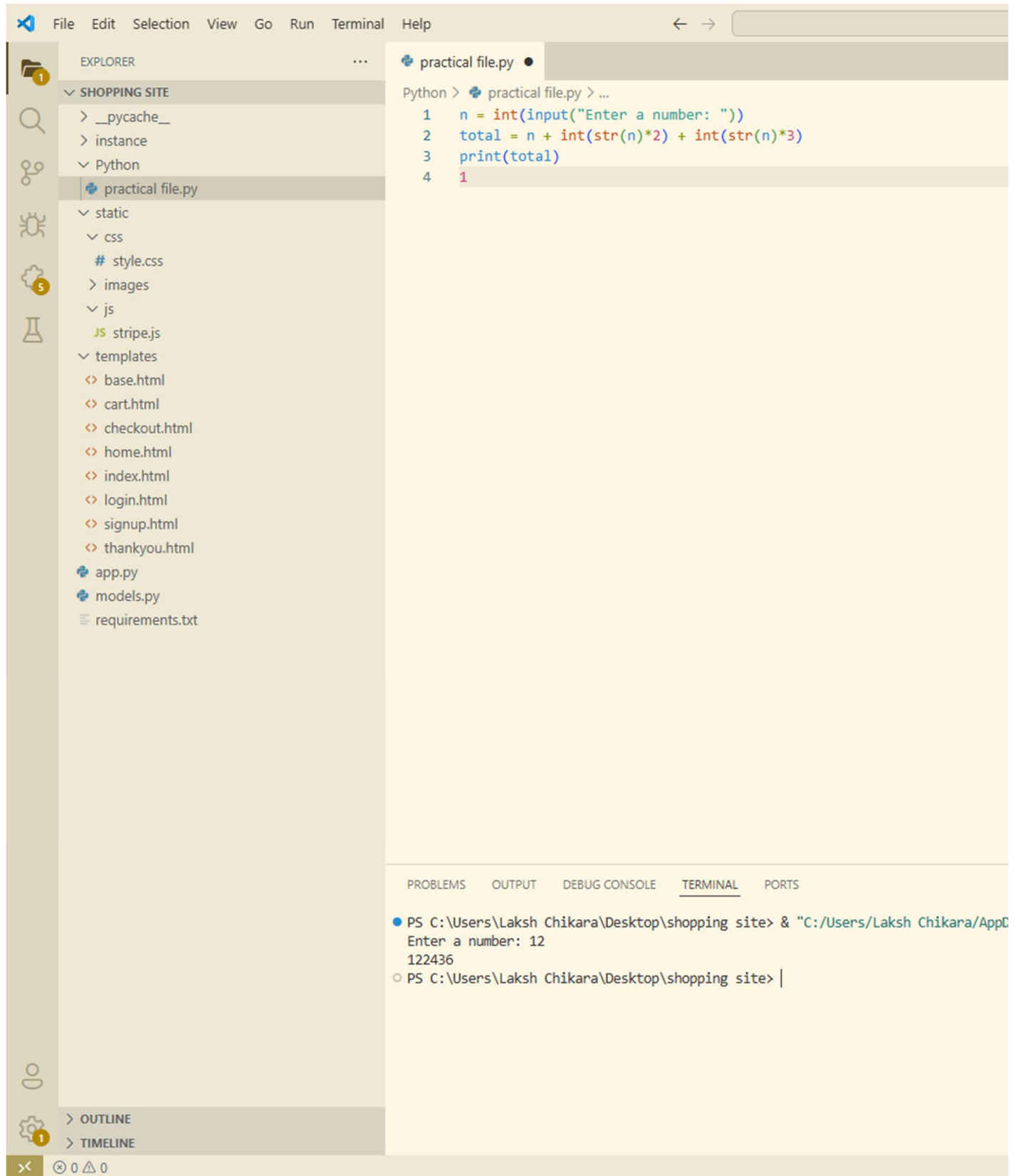
The screenshot shows a Visual Studio Code editor window with a project named "SHOPPING SITE". The Explorer sidebar on the left displays the project structure, including folders like __pycache__, instance, Python, static, css, images, js, and templates, along with files like app.py, models.py, and requirements.txt. The main editor area shows a file named "practical file.py" with the following Python code:

```
1 from datetime import date
2
3 date1 = date(2014, 7, 2)
4 date2 = date(2014, 7, 11)
5 delta = date2 - date1
6 print(f"{delta.days} days")
7
```

At the bottom of the window, the TERMINAL panel shows the execution output:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python38-32/Python.exe" "C:/Users/Laksh Chikara/Desktop/shopping site/practical file.py"
9 days
PS C:\Users\Laksh Chikara\Desktop\shopping site>
```

Ques:2) Write a Python program that accepts an integer (n) and computes the value of $n+nn+nnn$.



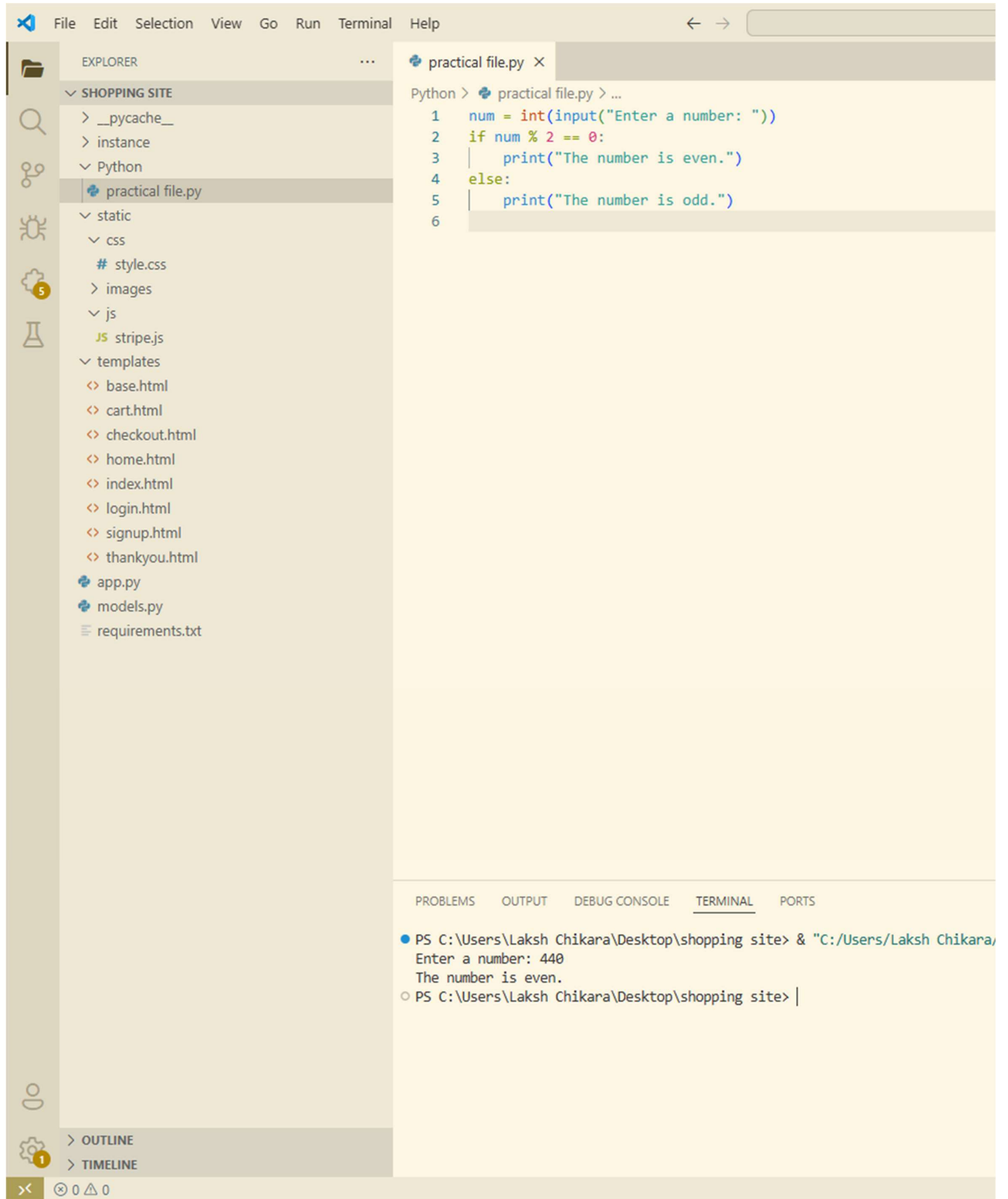
The screenshot shows a Visual Studio Code editor with a project named "SHOPPING SITE". The Explorer sidebar on the left lists the project structure, including files like `__pycache__`, `instance`, `Python`, `practical file.py`, `static`, `css`, `style.css`, `images`, `js`, `stripe.js`, `templates`, `base.html`, `cart.html`, `checkout.html`, `home.html`, `index.html`, `login.html`, `signup.html`, `thankyou.html`, `app.py`, `models.py`, and `requirements.txt`. The main editor area displays the file `practical file.py` with the following Python code:

```
1 n = int(input("Enter a number: "))
2 total = n + int(str(n)*2) + int(str(n)*3)
3 print(total)
4 1
```

At the bottom, the TERMINAL panel shows the command prompt output:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppD
Enter a number: 12
122436
PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:3) Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. Hint: how does an even / odd number react differently when divided by 2?



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project structure for 'SHOPPING SITE' with folders like __pycache__, instance, and Python, and files like practical file.py. The main editor window shows the code for 'practical file.py'.

```
1 num = int(input("Enter a number: "))
2 if num % 2 == 0:
3     print("The number is even.")
4 else:
5     print("The number is odd.")
6
```

The TERMINAL panel at the bottom shows the command prompt output:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/
Enter a number: 440
The number is even.
PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:4) Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.

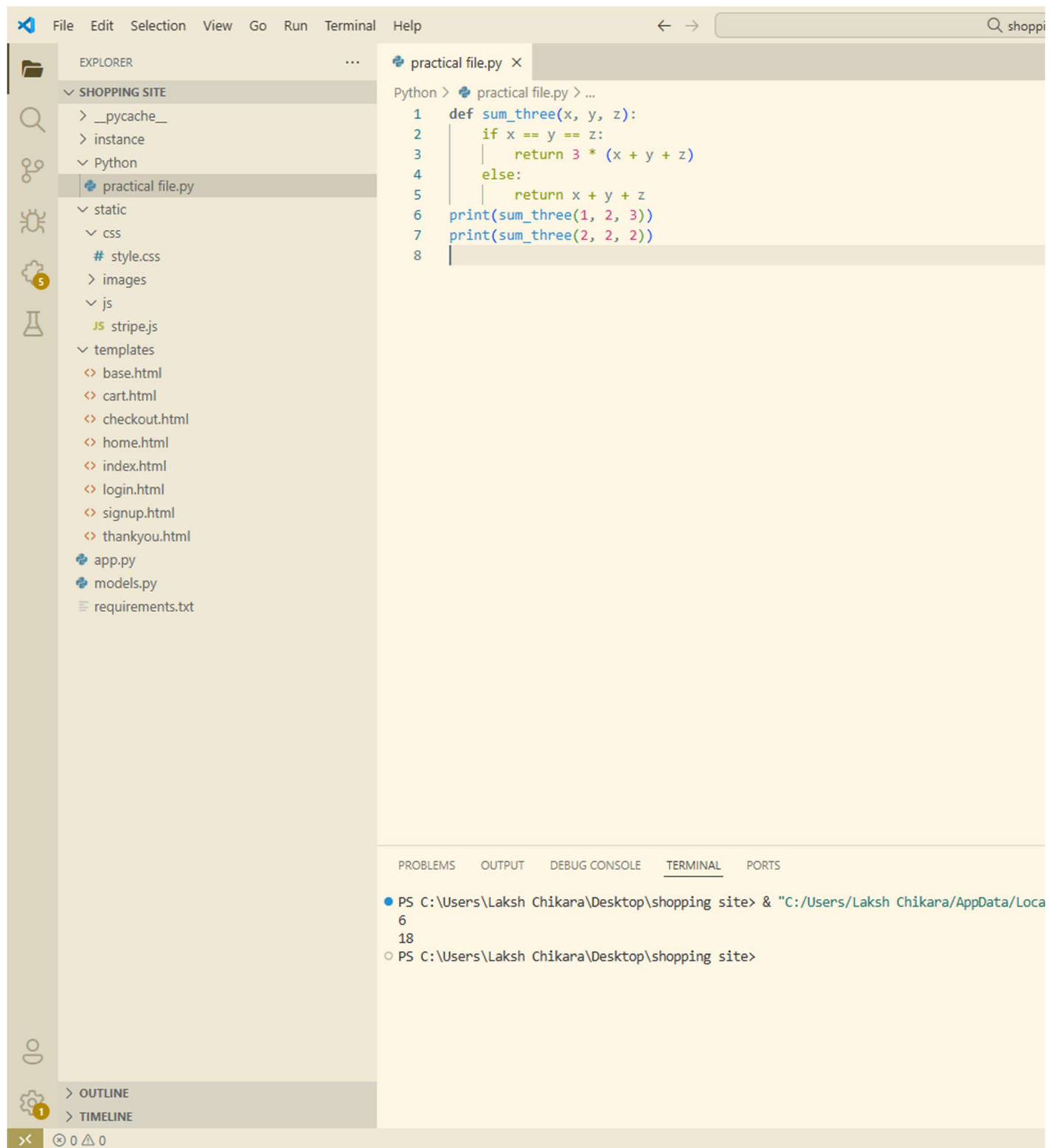
The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'SHOPPING SITE' with various files including HTML templates, JavaScript, CSS, and Python files. The main editor window is open to 'practical file.py', which contains the following Python code:

```
1 values = input("Enter comma-separated numbers: ")
2 list_values = values.split(',')
3 tuple_values = tuple(list_values)
4 print(list_values)
5 print(tuple_values)
6
```

At the bottom, the TERMINAL panel shows the execution of the program. The prompt is 'PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/...'. The user input is '10,20,30,40,50'. The program outputs the list ['10', '20', '30', '40', '50'] and the tuple ('10', '20', '30', '40', '50').

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/...
Enter comma-separated numbers: 10,20,30,40,50
['10', '20', '30', '40', '50']
('10', '20', '30', '40', '50')
PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:5) Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum.



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'SHOPPING SITE' with various files and folders. The code editor shows a Python file named 'practical file.py' with the following code:

```
1 def sum_three(x, y, z):
2     if x == y == z:
3         return 3 * (x + y + z)
4     else:
5         return x + y + z
6 print(sum_three(1, 2, 3))
7 print(sum_three(2, 2, 2))
8
```

The terminal at the bottom shows the execution of the program:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python39-64/Python.exe" "C:/Users/Laksh Chikara/Desktop/shopping site/practical file.py"
6
18
PS C:\Users\Laksh Chikara\Desktop\shopping site>
```

Ques:6) Write a Python program to test whether a passed letter is a vowel or not

SHOPPING SITE

> __pycache__

> instance

Python

practical file.py

static

css

style.css

images

js

JS stripe.js

templates

<> base.html

<> cart.html

<> checkout.html

<> home.html

<> index.html

<> login.html

<> signup.html

<> thankyou.html

app.py

models.py

requirements.txt

Python > practical file.py > ...

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

2 if letter in 'aeiou':

3 | print("Vowel")

4 else:

5 | print("Not a vowel")

6

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/

Enter a letter: I

Vowel

○ PS C:\Users\Laksh Chikara\Desktop\shopping site> |

> OUTLINE

> TIMELINE

0 0 0

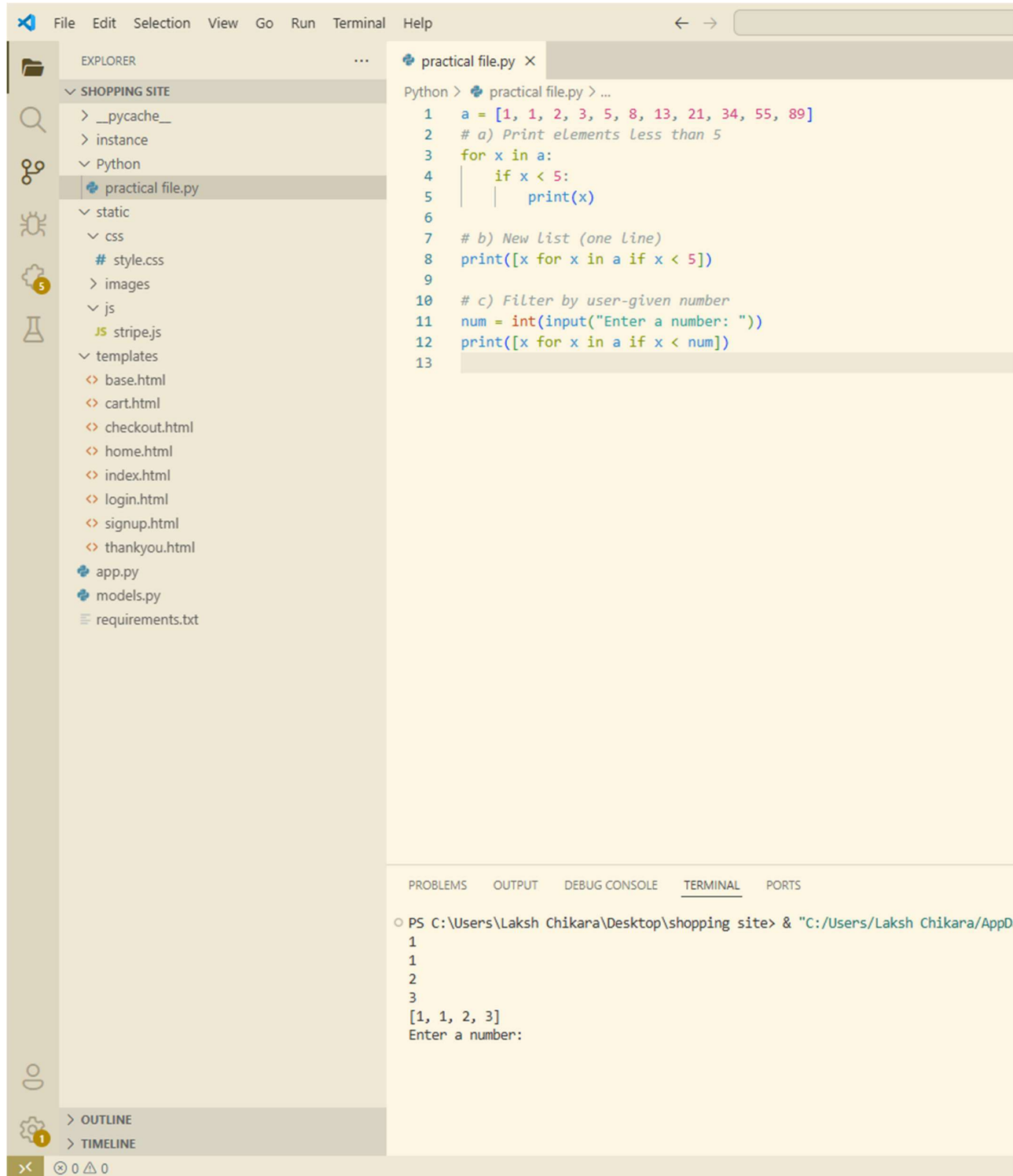
Ques:7) Take a list, say for example this one:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

and write a program that prints out all the elements of the list that are less than 5.

Extras:

- Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.**
- Write this in one line of Python.**
- Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.**



The screenshot shows a Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'SHOPPING SITE' with various files and folders. The code editor shows a file named 'practical file.py' with the following Python code:

```
1 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
2 # a) Print elements less than 5
3 for x in a:
4     if x < 5:
5         print(x)
6
7 # b) New List (one line)
8 print([x for x in a if x < 5])
9
10 # c) Filter by user-given number
11 num = int(input("Enter a number: "))
12 print([x for x in a if x < num])
13
```

The bottom panel shows the 'TERMINAL' output, which displays the execution of the script. It shows the list elements less than 5, the new list, and the prompt for a user-given number.

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppD
1
1
2
3
[1, 1, 2, 3]
Enter a number:
```

Ques:8) Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because $26 / 13$ has no remainder.)

The screenshot shows a Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'SHOPPING SITE' with various files including HTML templates, CSS, JavaScript, and Python files. The file 'practical file.py' is selected and open in the main editor. The code in the editor is a Python script that prompts the user for a number and prints its divisors. The terminal at the bottom shows the command prompt where the program was executed, with the input '07' and the output '[1, 7]'.

```
practical file.py
Python > practical file.py > ...
1 num = int(input("Enter a number: "))
2 divisors = []
3 for i in range(1, num + 1):
4     if num % i == 0:
5         divisors.append(i)
6 print(divisors)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

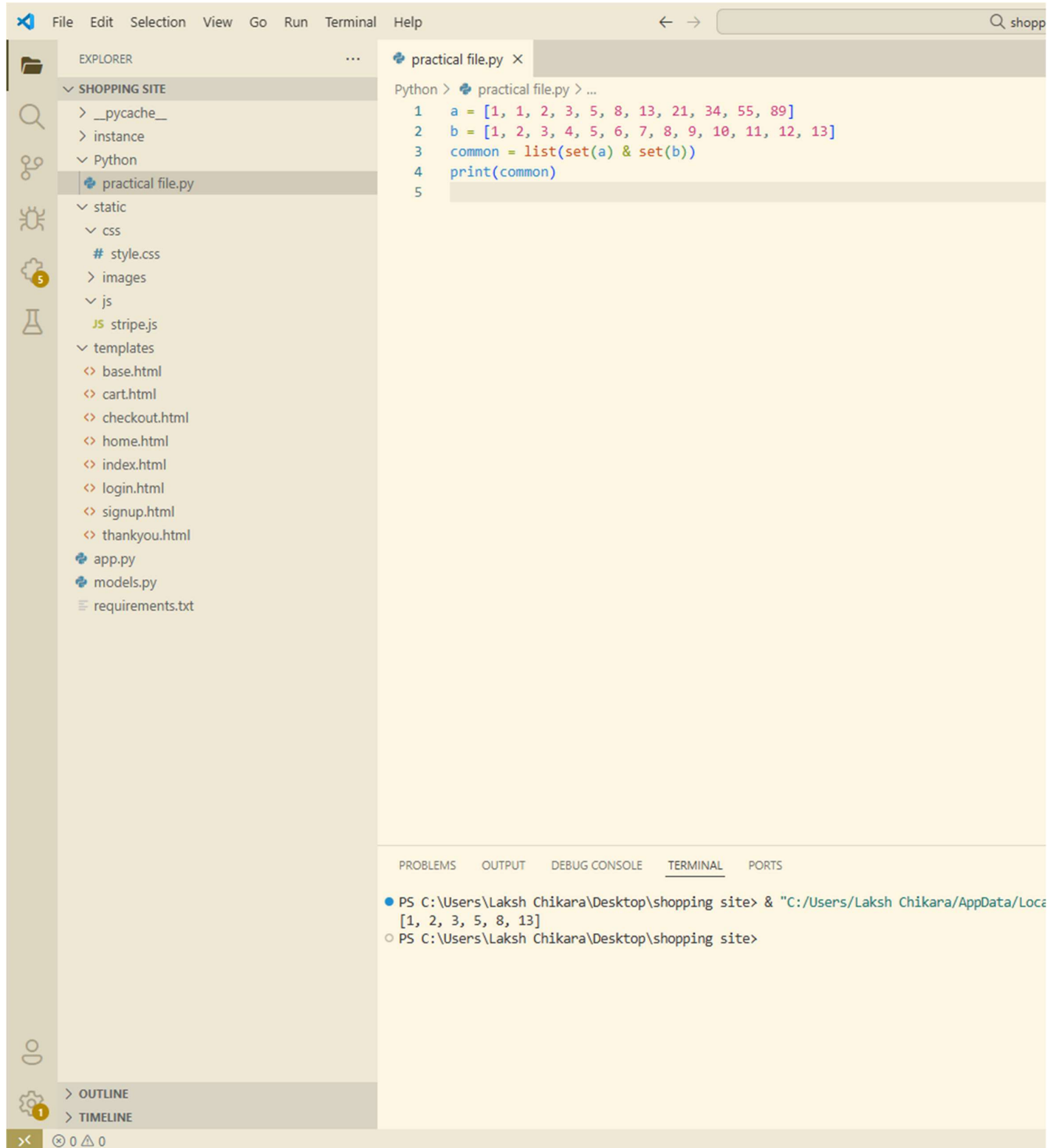
```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/Py
Enter a number: 07
[1, 7]
PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:9) Take two lists, say for example these two:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.



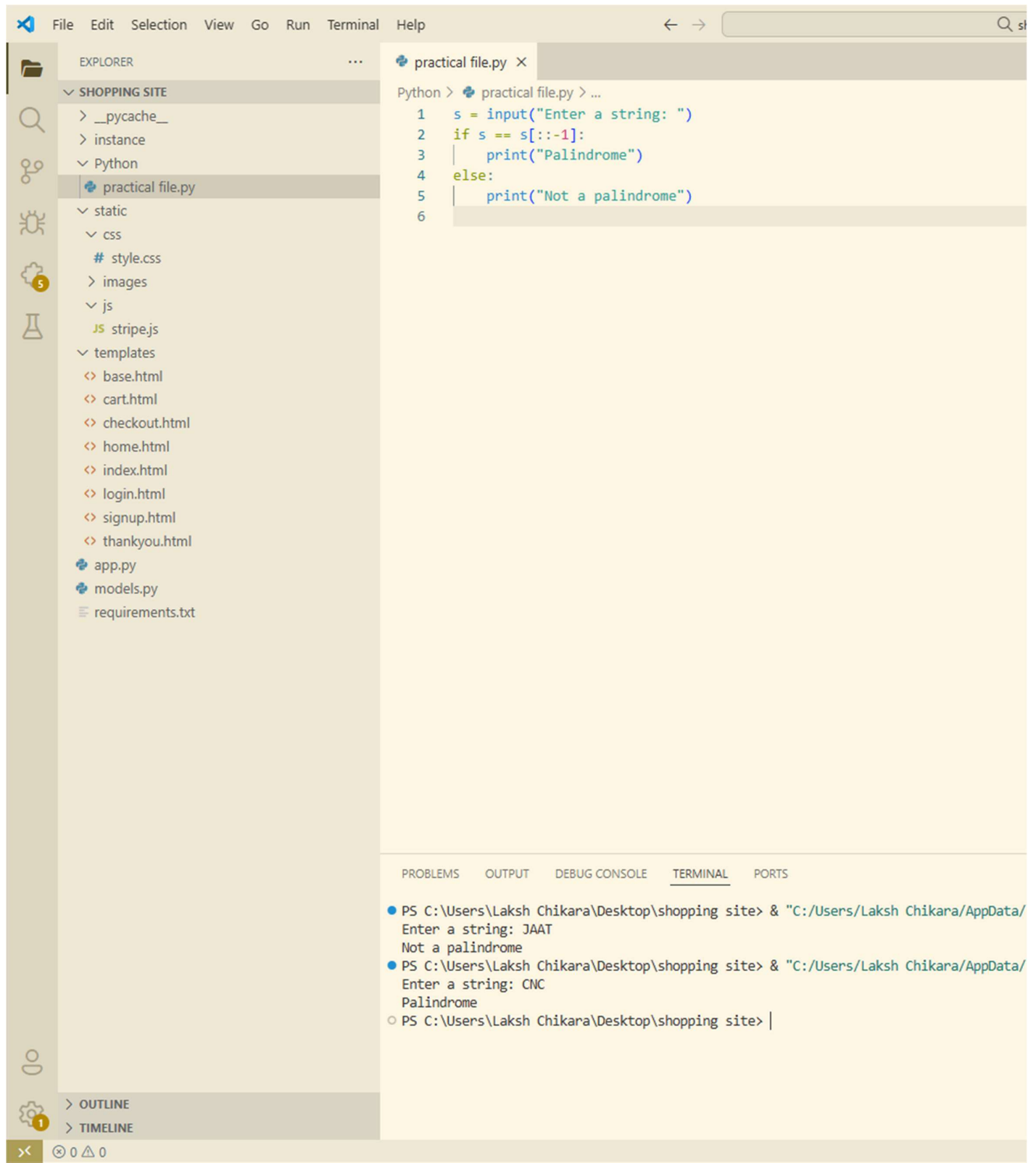
The screenshot shows a Visual Studio Code editor window. The Explorer sidebar on the left displays a project structure for 'SHOPPING SITE' with folders like __pycache__, instance, Python, static, css, images, js, templates, and files like app.py, models.py, and requirements.txt. The main editor area shows a file named 'practical file.py' with the following Python code:

```
1 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
2 b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
3 common = list(set(a) & set(b))
4 print(common)
5
```

Below the editor, the TERMINAL panel shows the command prompt output:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python39-64/Python.exe" practical file.py
[1, 2, 3, 5, 8, 13]
PS C:\Users\Laksh Chikara\Desktop\shopping site>
```

Ques:10) Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)



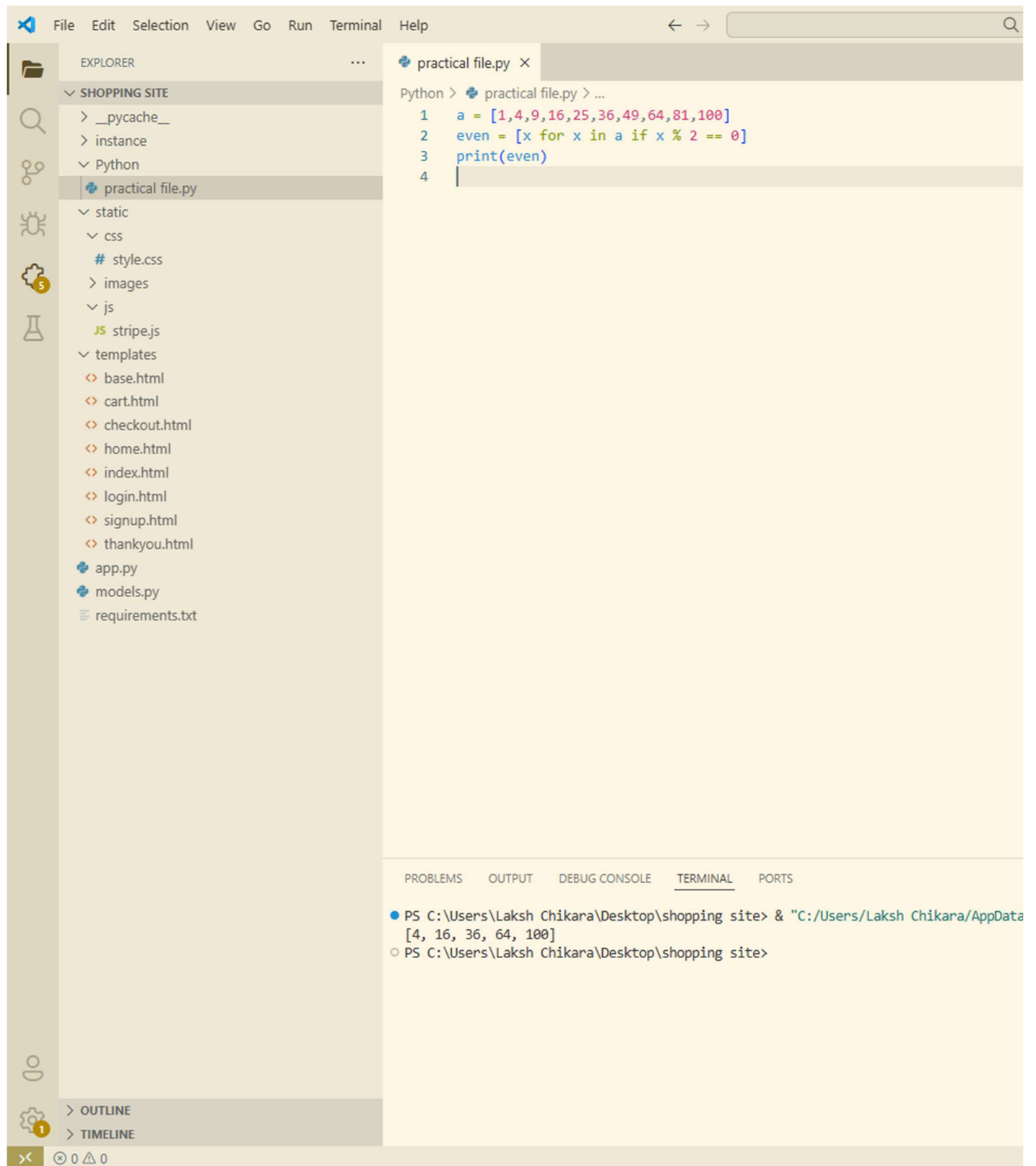
The screenshot shows a Visual Studio Code editor window. On the left, the Explorer sidebar displays a project named 'SHOPPING SITE' with various files including HTML templates, CSS, JS, and Python files. The main editor area shows a file named 'practical file.py' containing a Python script. The script prompts the user to 'Enter a string:' and checks if the string is a palindrome by comparing it to its reverse. The script uses the slicing technique `s[::-1]`. Below the editor, the TERMINAL panel shows the execution of the script. It displays two test cases: 'JAAT' which is correctly identified as 'Not a palindrome', and 'CNC' which is correctly identified as 'Palindrome'. The terminal prompt is `PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/`.

```
practical file.py X
Python > practical file.py > ...
1 s = input("Enter a string: ")
2 if s == s[::-1]:
3     print("Palindrome")
4 else:
5     print("Not a palindrome")
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/
Enter a string: JAAT
Not a palindrome
- PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/
Enter a string: CNC
Palindrome
- PS C:\Users\Laksh Chikara\Desktop\shopping site> |

Ques:11) Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project structure for 'SHOPPING SITE' with folders like __pycache__, instance, and Python, and files like practical file.py. The main editor window shows the code in 'practical file.py':

```
1 a = [1,4,9,16,25,36,49,64,81,100]
2 even = [x for x in a if x % 2 == 0]
3 print(even)
4
```

The TERMINAL panel at the bottom shows the command prompt output:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData
[4, 16, 36, 64, 100]
PS C:\Users\Laksh Chikara\Desktop\shopping site>
```