AI ASSISTED CODING

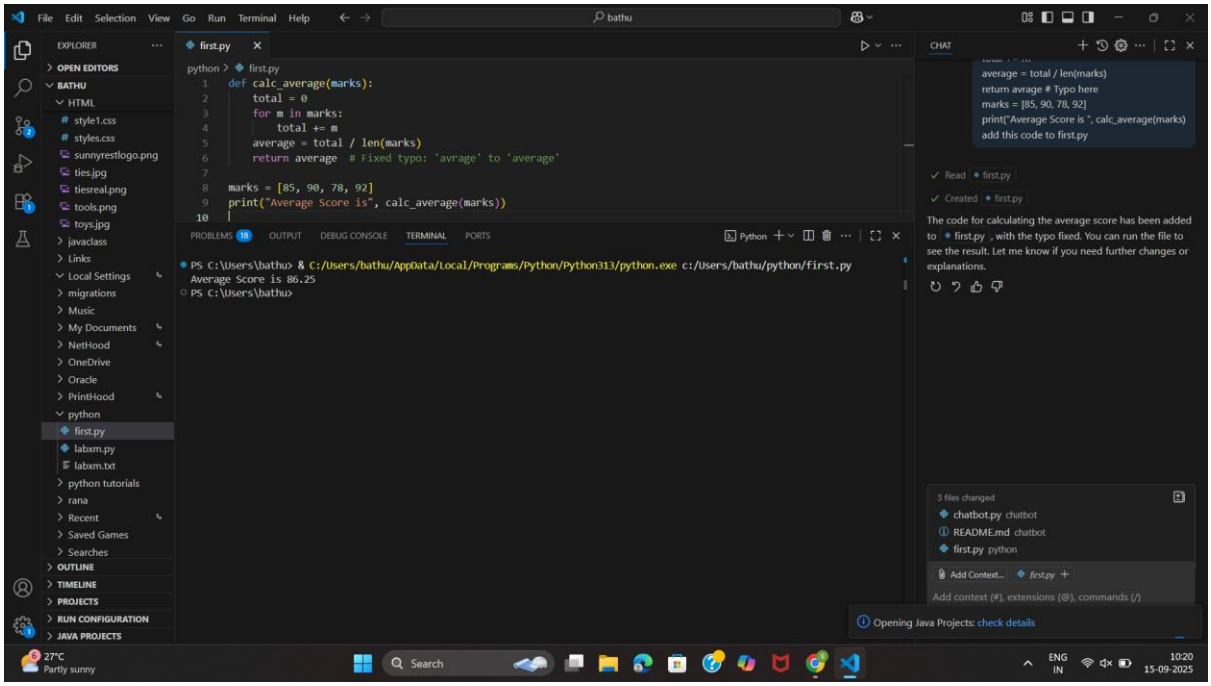
ASSIGNMENT – 10.1

2403A52187

BATCH – 18

Task 1: Use AI to identify and fix syntax and logic errors in a faulty

```
def calc_average(marks):
total = 0
for m in marks:
total += m
average = total / len(marks)
return avrage # Typo here
marks = [85, 90, 78, 92]
print("Average Score is ", calc_average(marks)
```



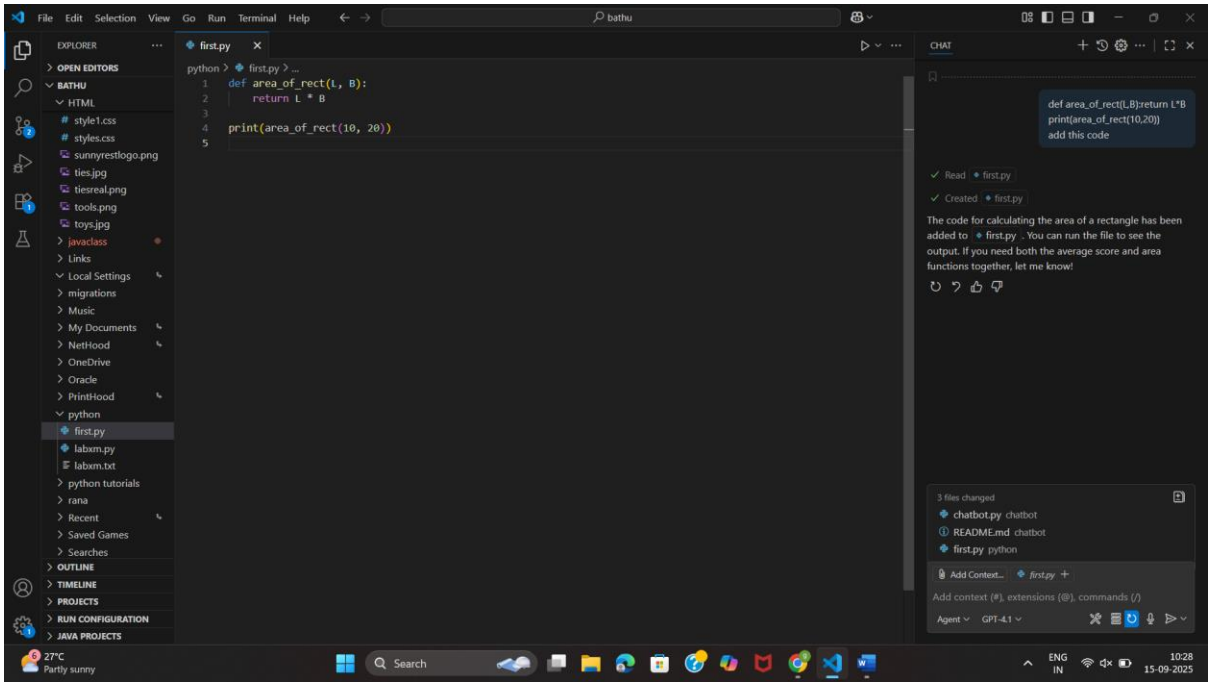
AI fixing the typo error.

Task 2: Use AI to refactor Python code to follow PEP 8 style guidelines.

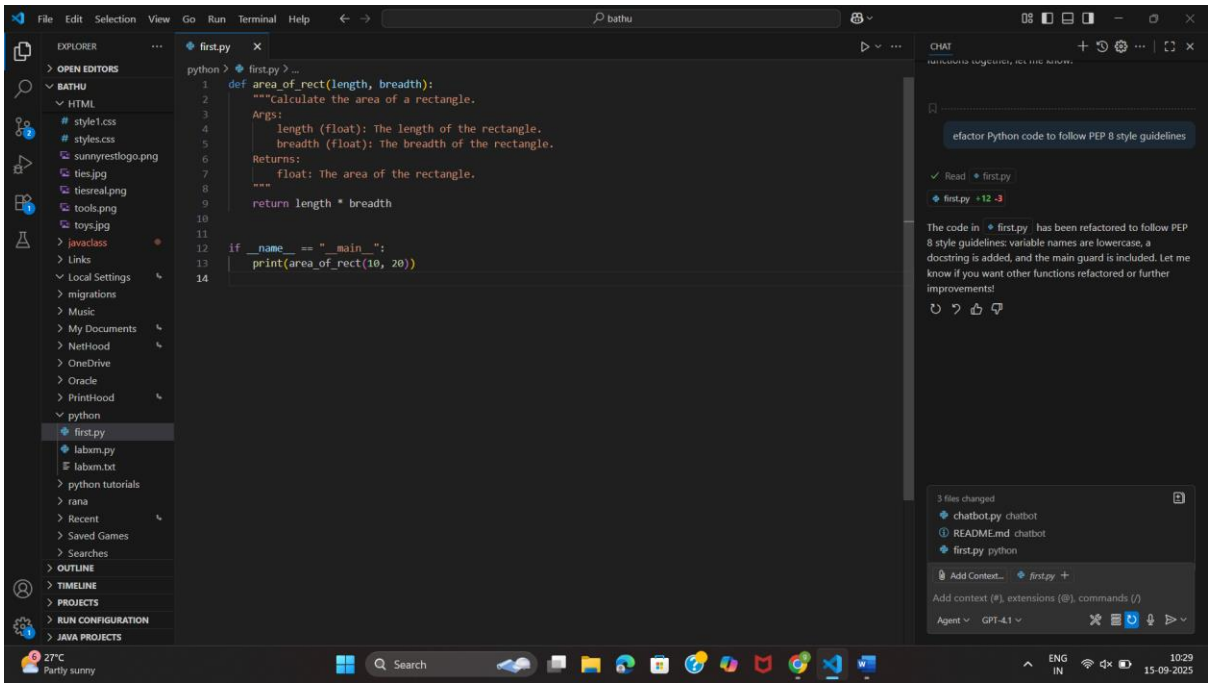
Sample Input Code:

def area_of_rect(L,B):return L*B

print(area_of_rect(10,20))



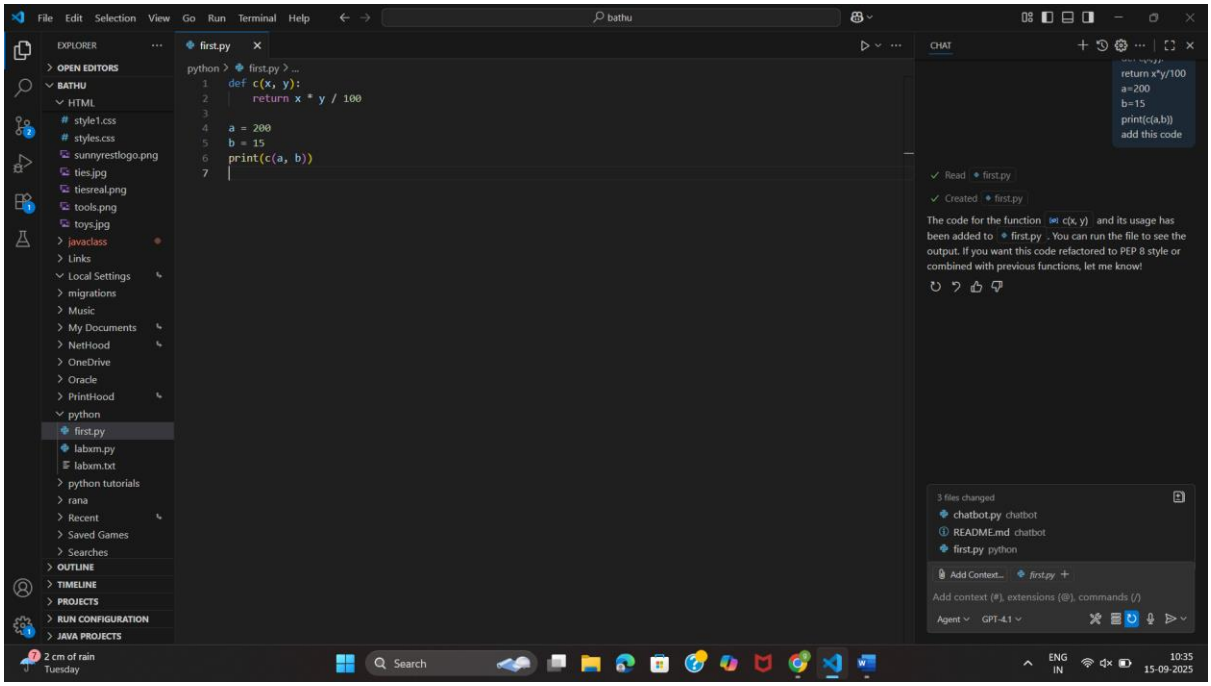
After following PEP 8 style guidelines



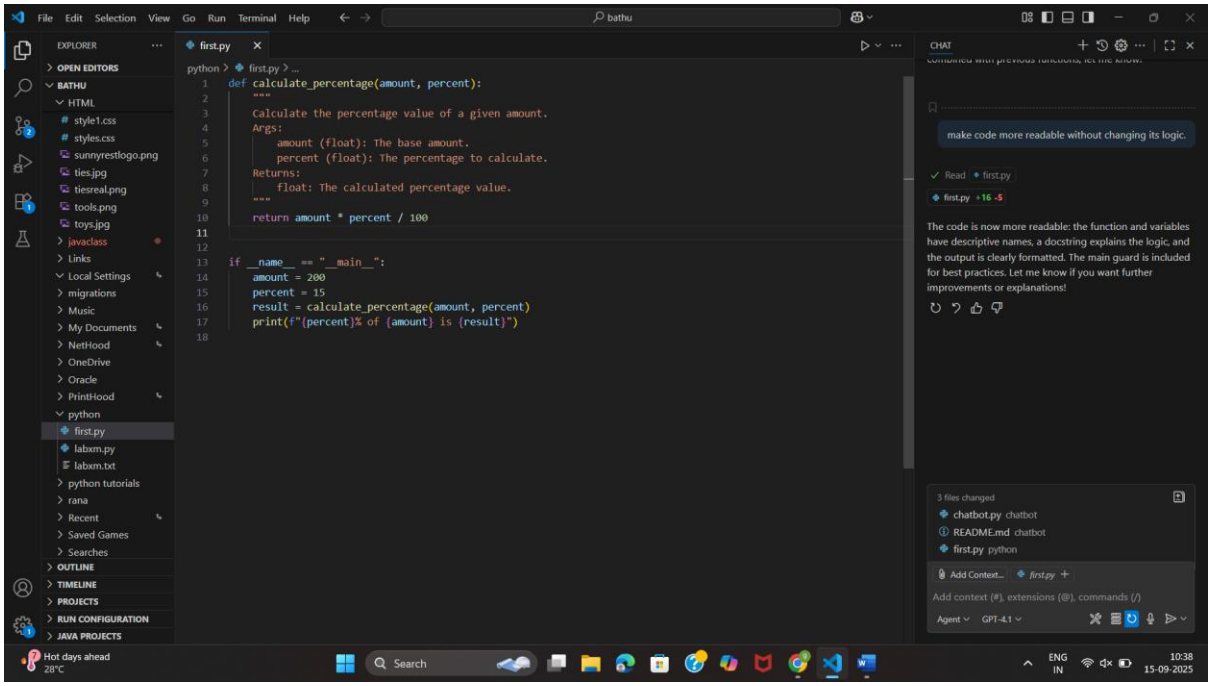
Task 3: Use AI to make code more readable without changing its logic.

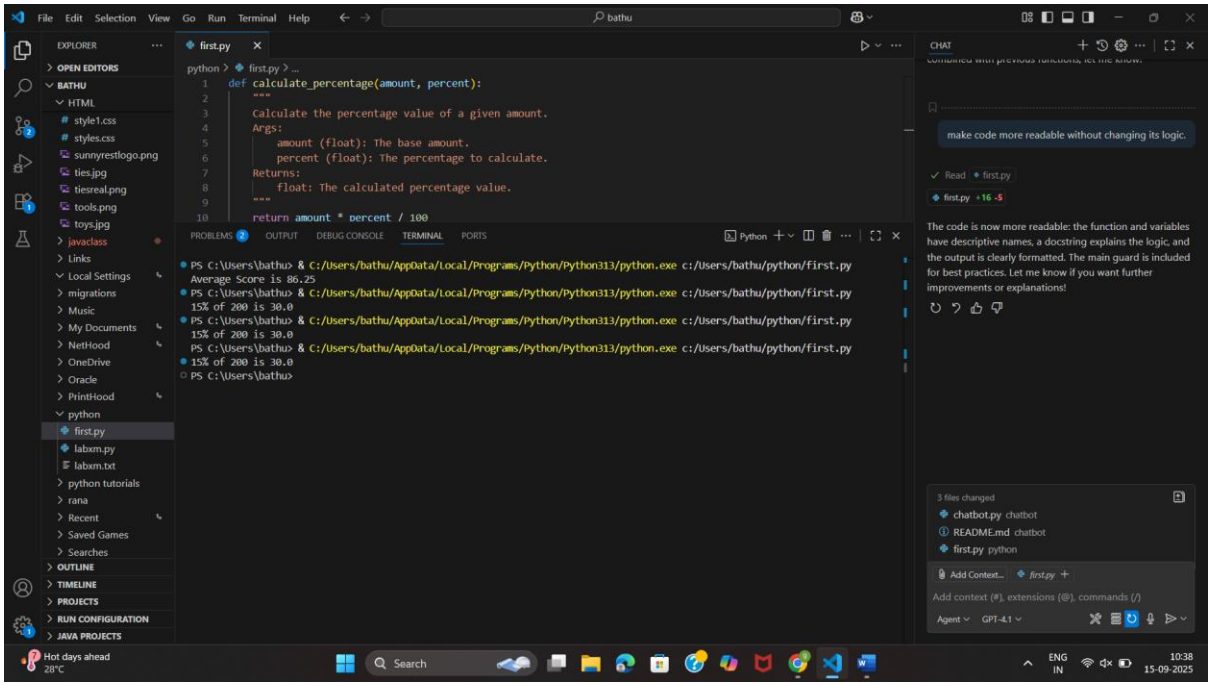
Sample Input Code:

```
def c(x,y):
return x*y/100
a=200
b=15
print(c(a,b))
```



Making code understandable without changing its logic





Task 4: Use AI to break repetitive or long code into reusable functions.

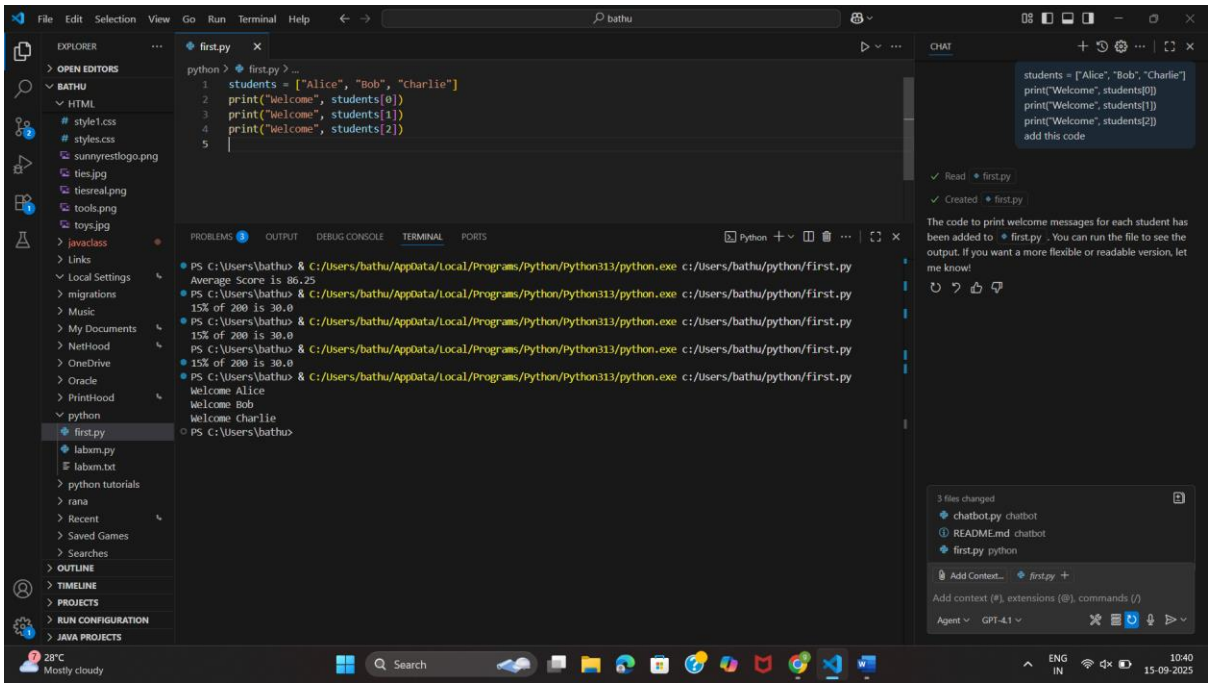
Sample Input Code:

students = ["Alice", "Bob", "Charlie"]

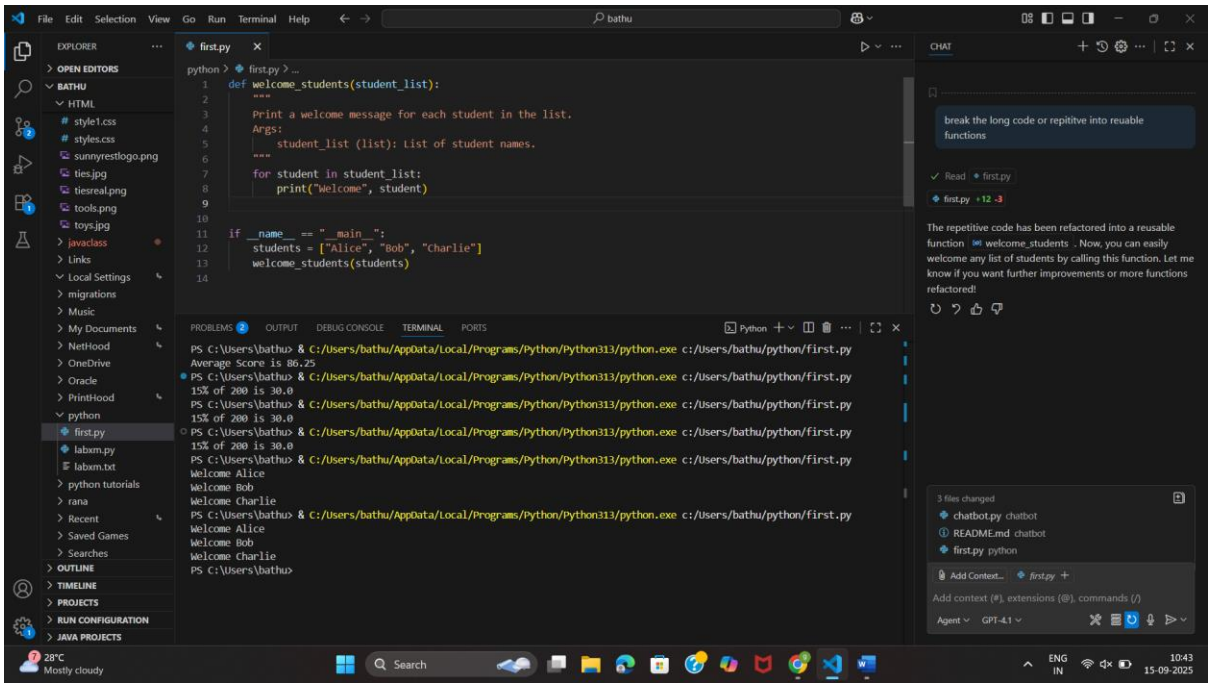
print("Welcome", students[0])

print("Welcome", students[1])

print("Welcome", students[2]



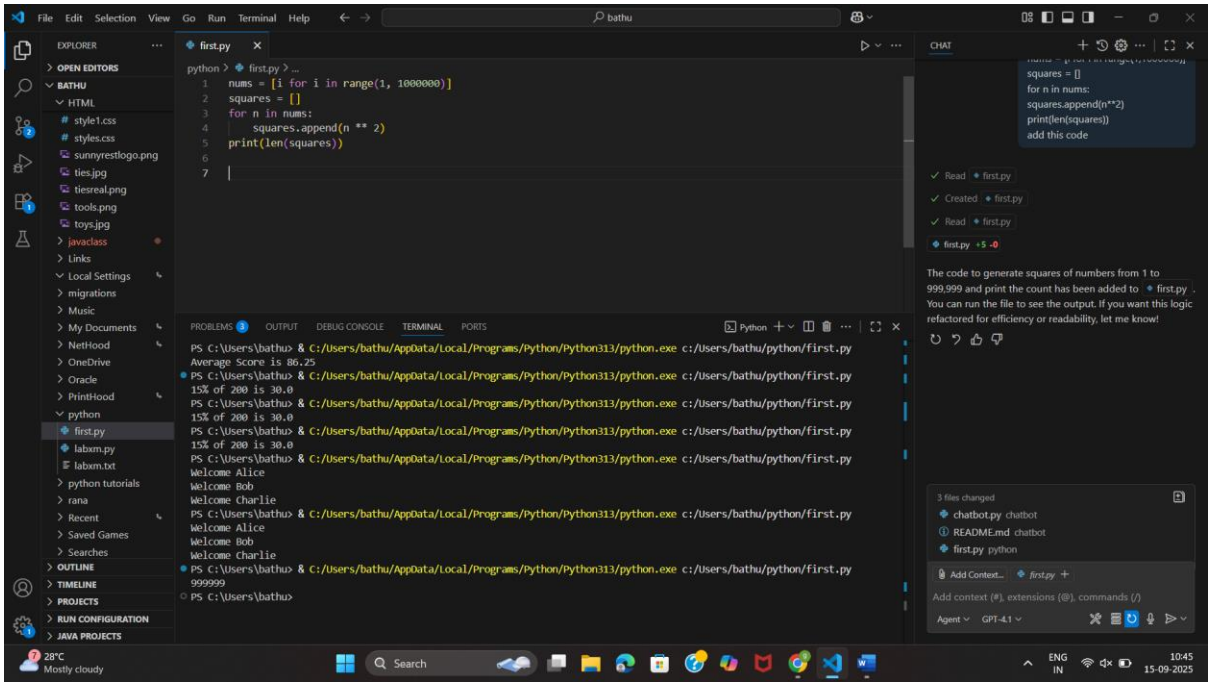
long code into reusable functions.



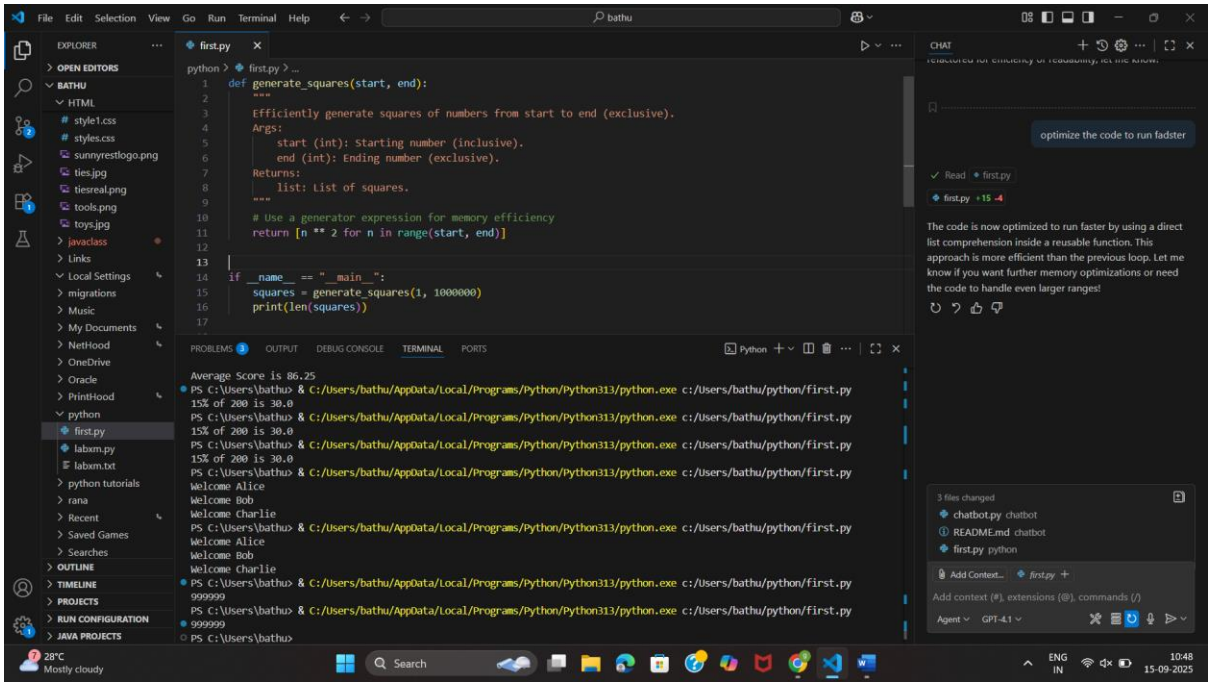
Task 5: Use AI to make the code run faster.

Sample Input Code:

```
# Find squares of numbers
nums = [i for i in range(1,1000000)]
squares = []
for n in nums:
squares.append(n**2)
print(len(squares))
```



Optimized code



Task 6: Use AI to simplify overly complex logic.

Sample Input Code:

```
def grade(score):
if score >= 90:
return "A"
```

else:
if score >= 80:
return "B"

else:
if score >= 70:
return "C"
else:
if score >= 60:
return "D"
else:
return "F"