***Employee-Wage-Computation***

**UC1: Added Employee Attendance Check**

print("Welcome to Employee Wage Computation Program")

import random

print("Welcome to Employee Wage Computation Program")

# UC1: Check Employee Attendance

attendance = random.choice([0, 1])

if attendance == 1:

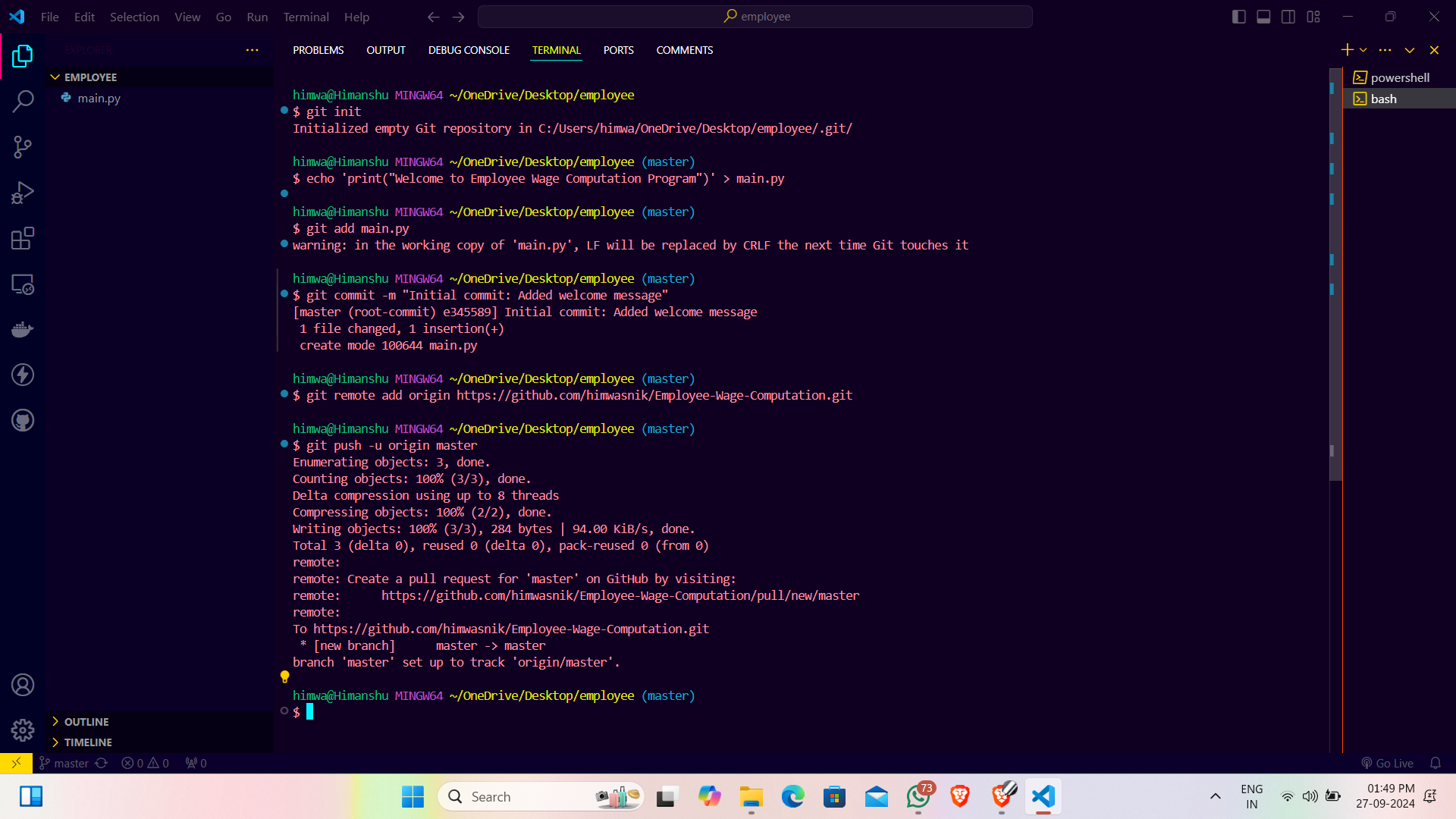
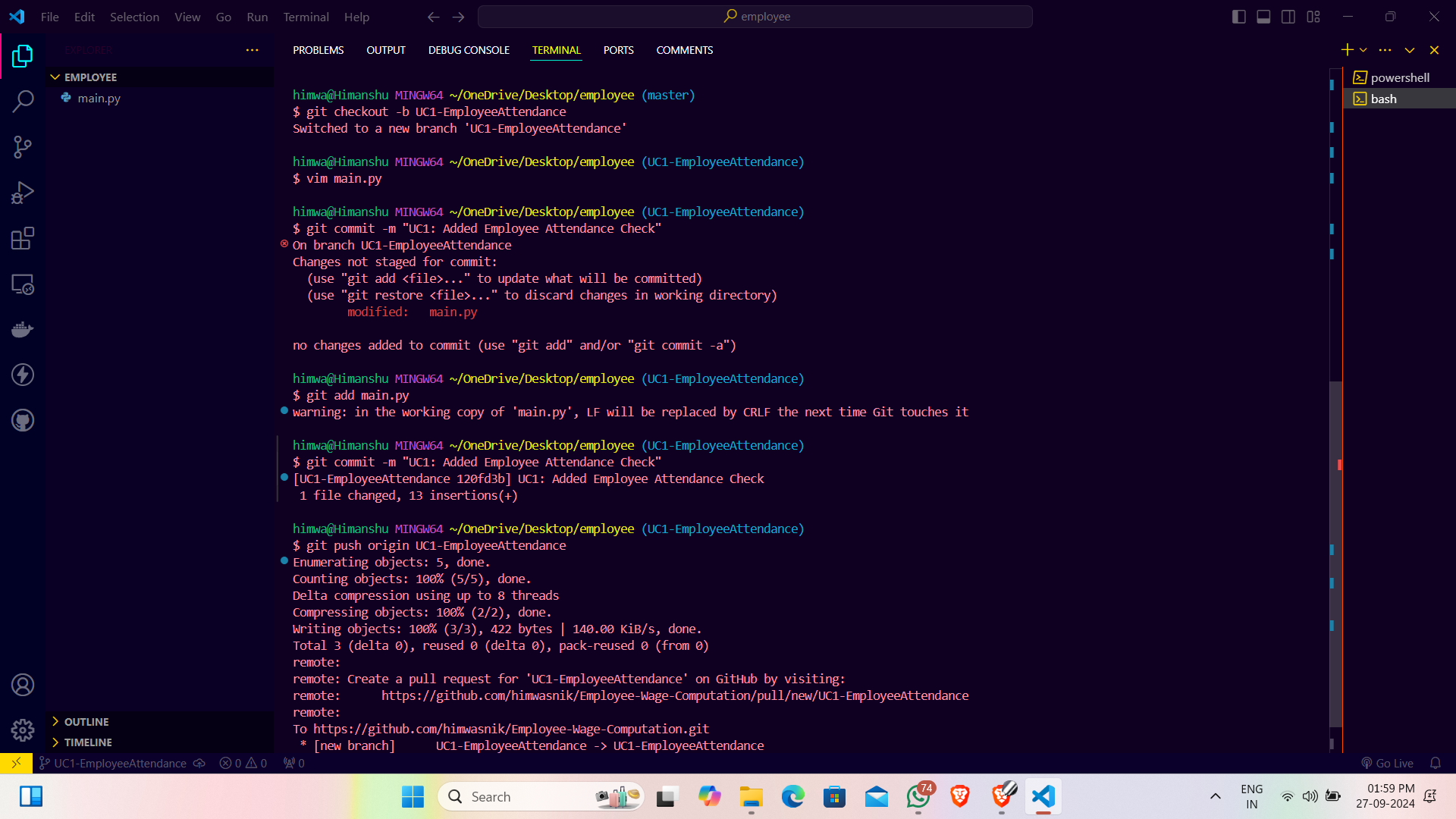
print("Employee is Present")

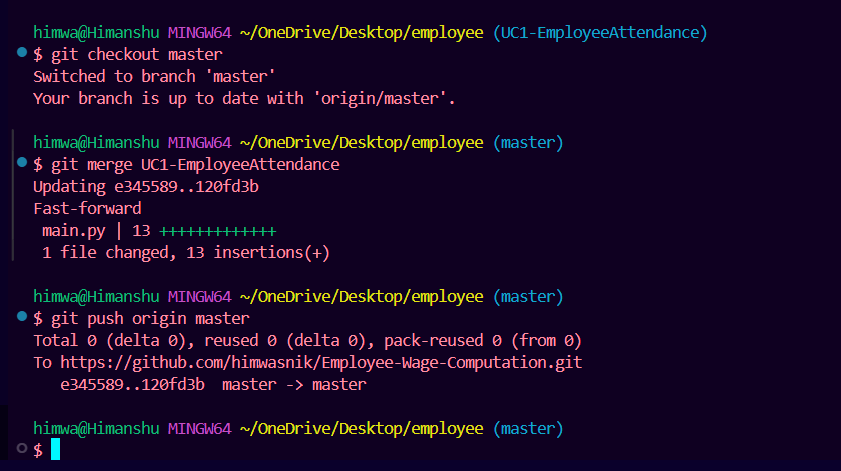
else:

print("Employee is Absent")

**Explanation:-**

* Class **Variables**: Constants such as WAGE\_PER\_HOUR, FULL\_DAY\_HOURS, PART\_TIME\_HOURS, etc., are class variables used throughout the calculations.
* Attendance **Check (UC 1)**: The is\_employee\_present() method uses random.choice() to randomly assign whether an employee is present or absent.





[***UC2: Added Daily Employee Wage Calculation***](https://github.com/himwasnik/Employee-Wage-Computation/commit/499065c06cf220ca12895d199b2ecf79a42a9dc9)

# UC2: Daily Employee Wage

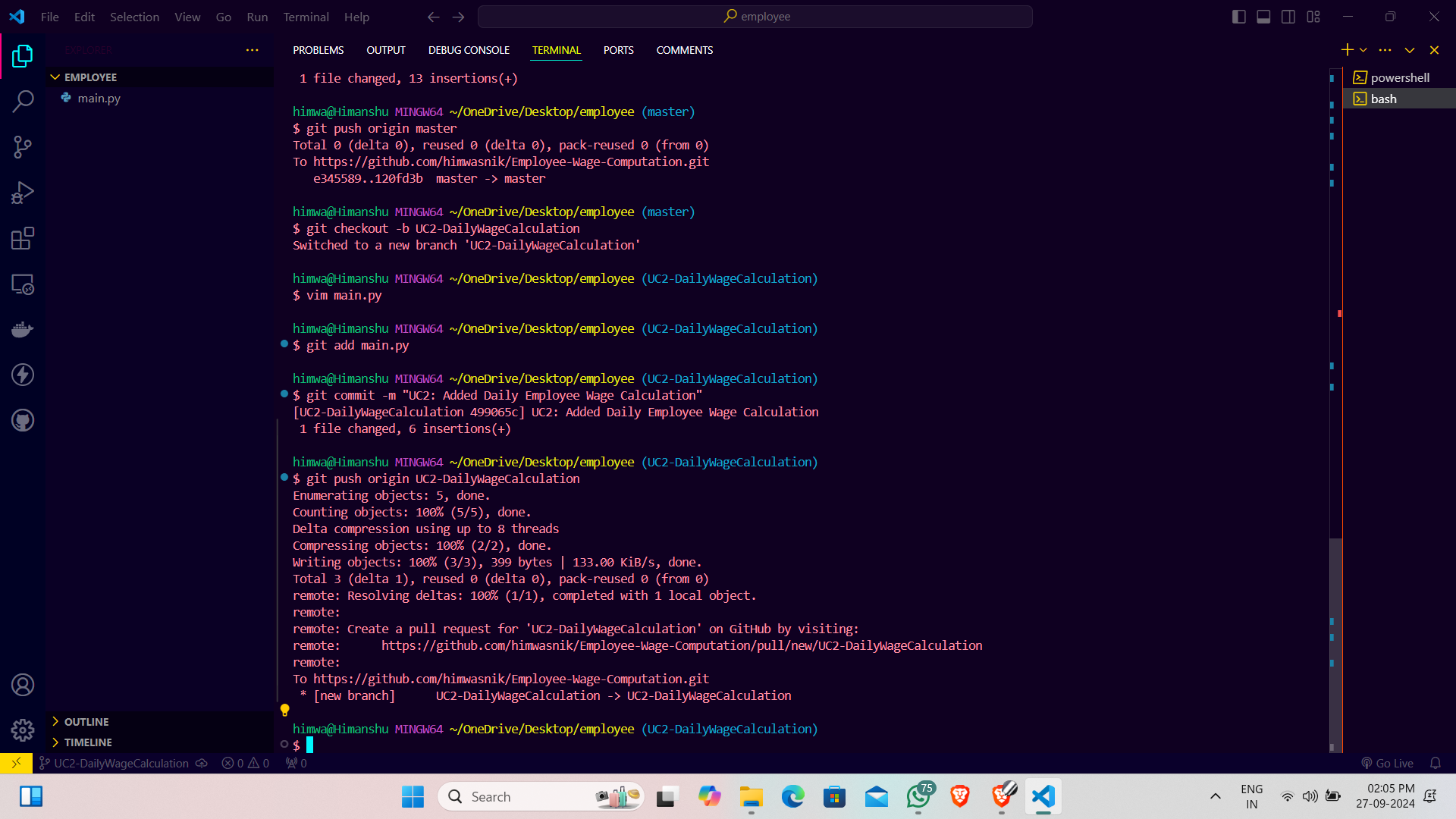
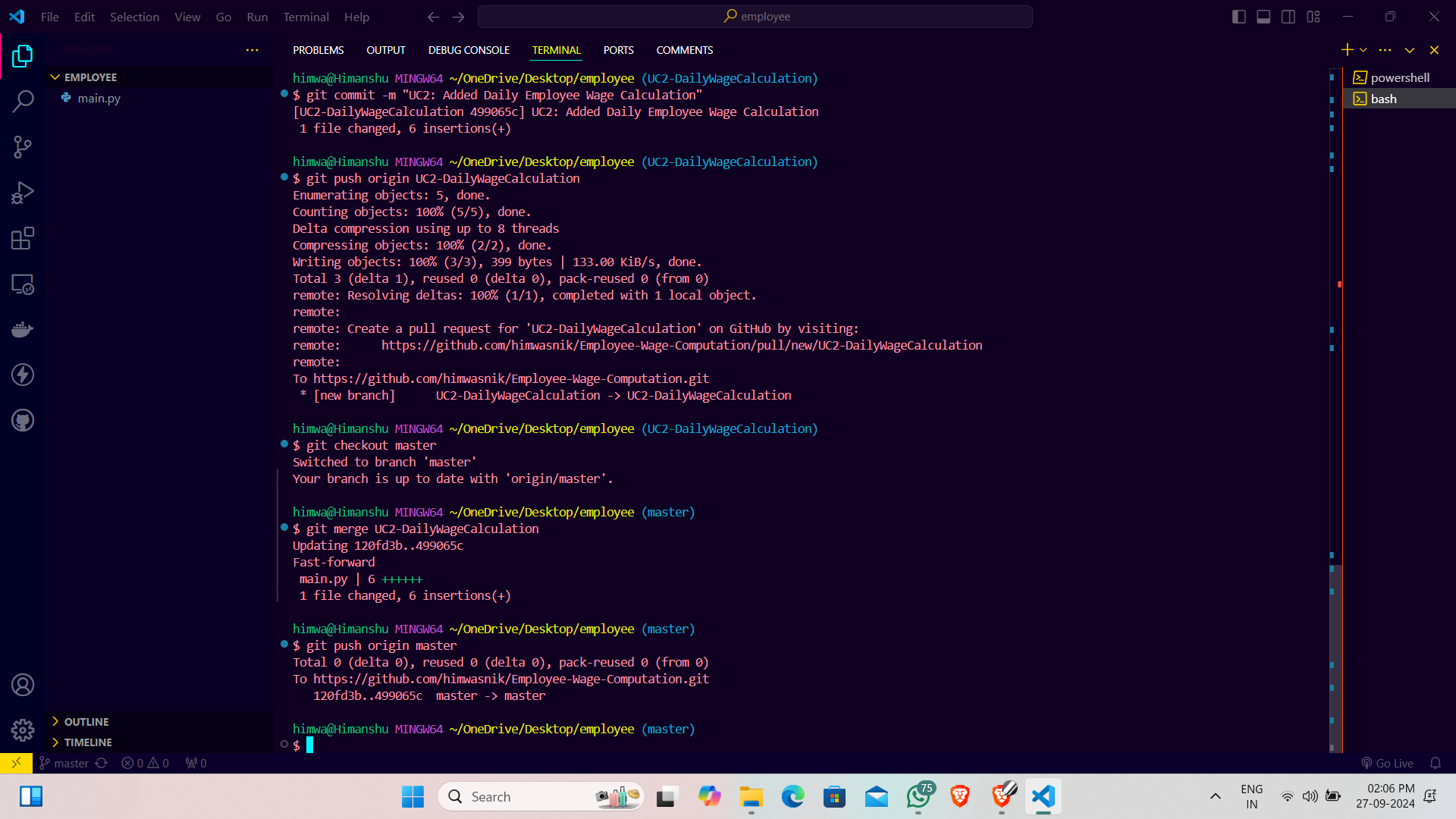
wage\_per\_hour = 20

full\_day\_hour = 8

daily\_wage = wage\_per\_hour \* full\_day\_hour

print(f"Daily Employee Wage: {daily\_wage}")

**Explanation:-**

Daily Wage Calculation (UC 2): calculate\_daily\_wage() takes the number of hours worked as input and calculates the daily wage based on WAGE\_PER\_HOUR

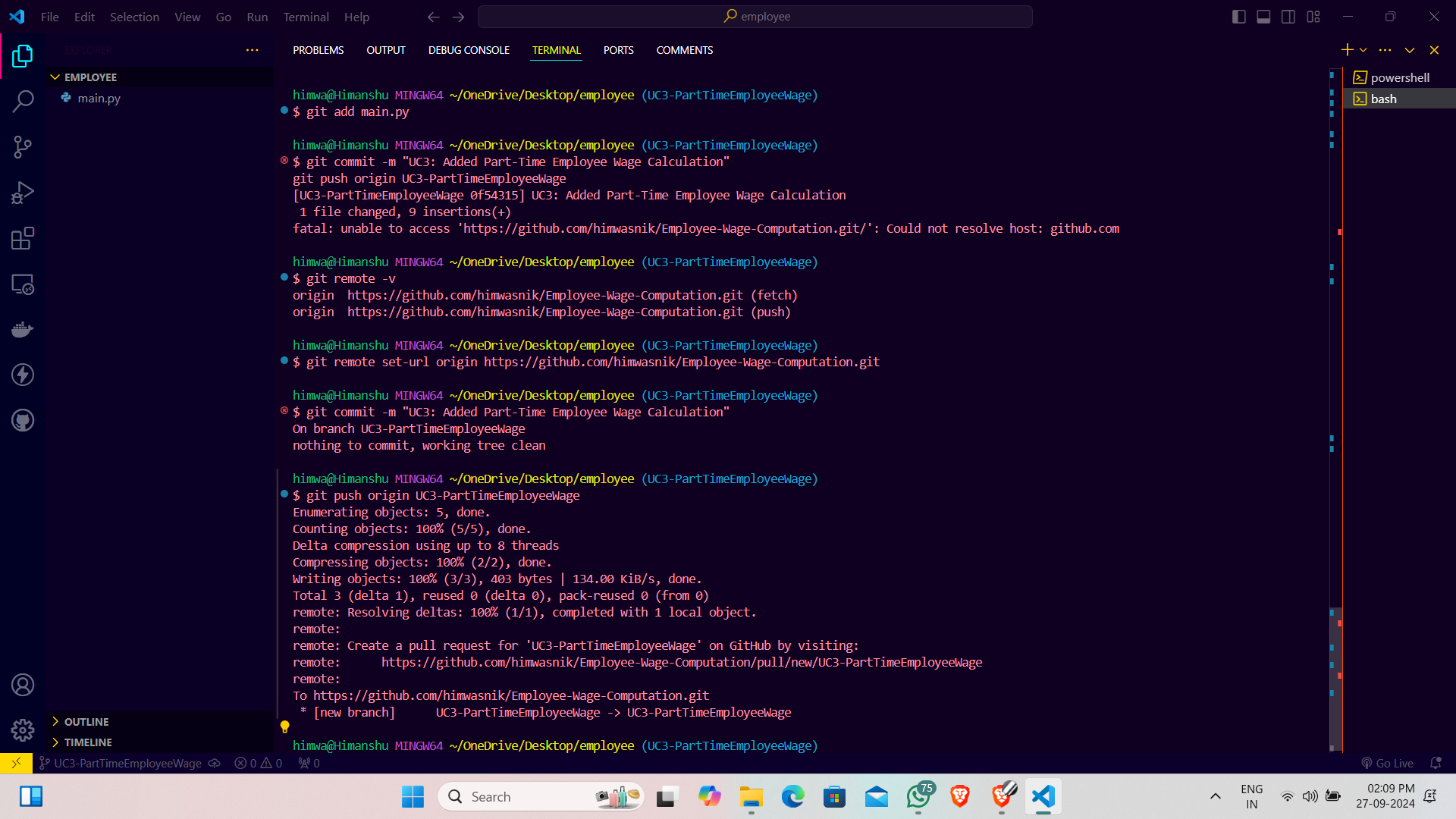
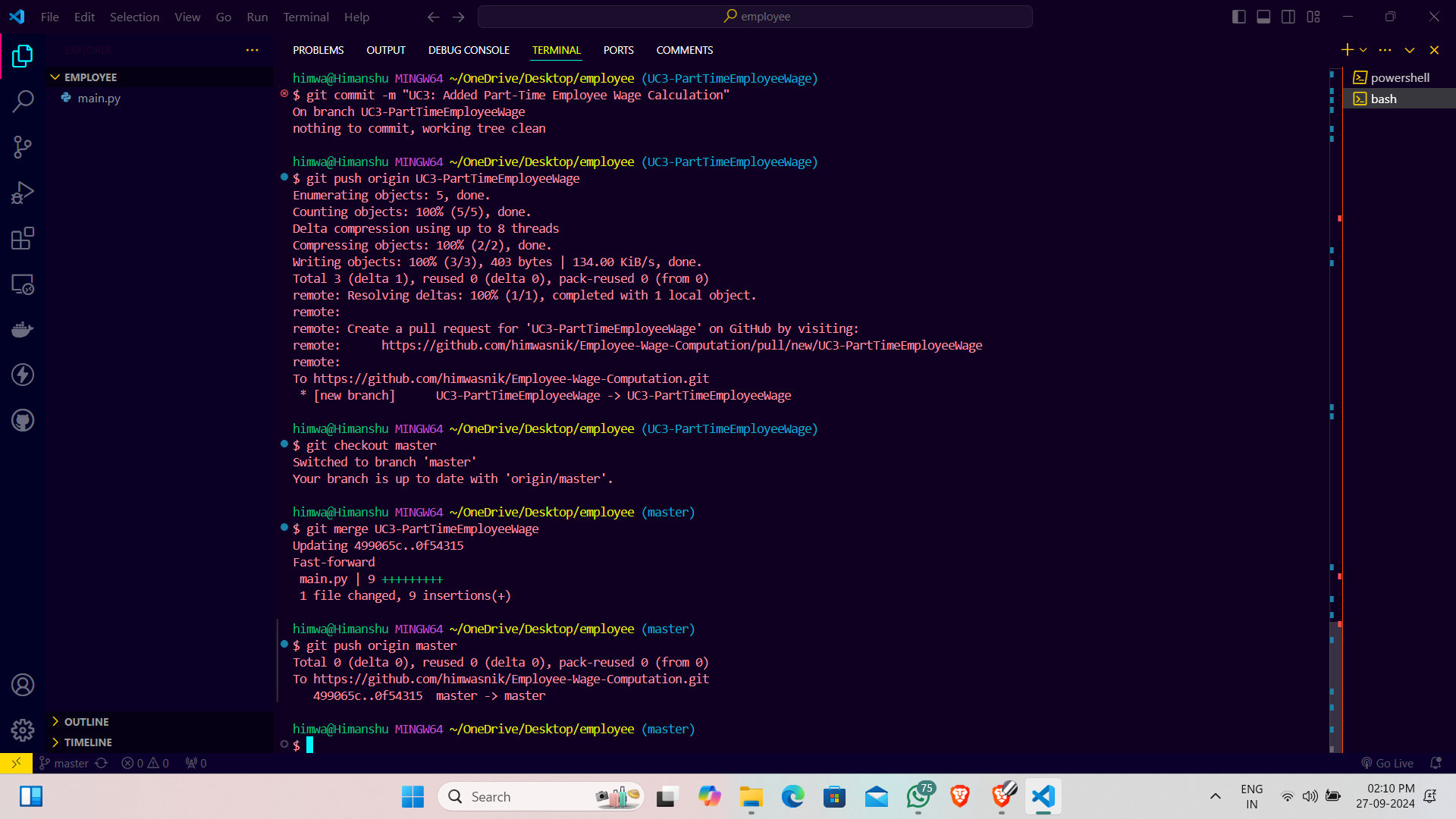
***UC3: Part-Time Employee Wage***

part\_time\_hour = 4

part\_time\_wage = wage\_per\_hour \* part\_time\_hour

print(f"Part-time Employee Wage: {part\_time\_wage}")

**Explanation:-**

Part-Time Wage (UC 3): part\_time\_employee\_wage() method calculates the wage based on part-time hours.

***UC4: Solve using Switch Case***

def employee\_wage(employee\_type):

switcher = {

1: full\_day\_hour \* wage\_per\_hour, # Full-time employee

2: part\_time\_hour \* wage\_per\_hour # Part-time employee

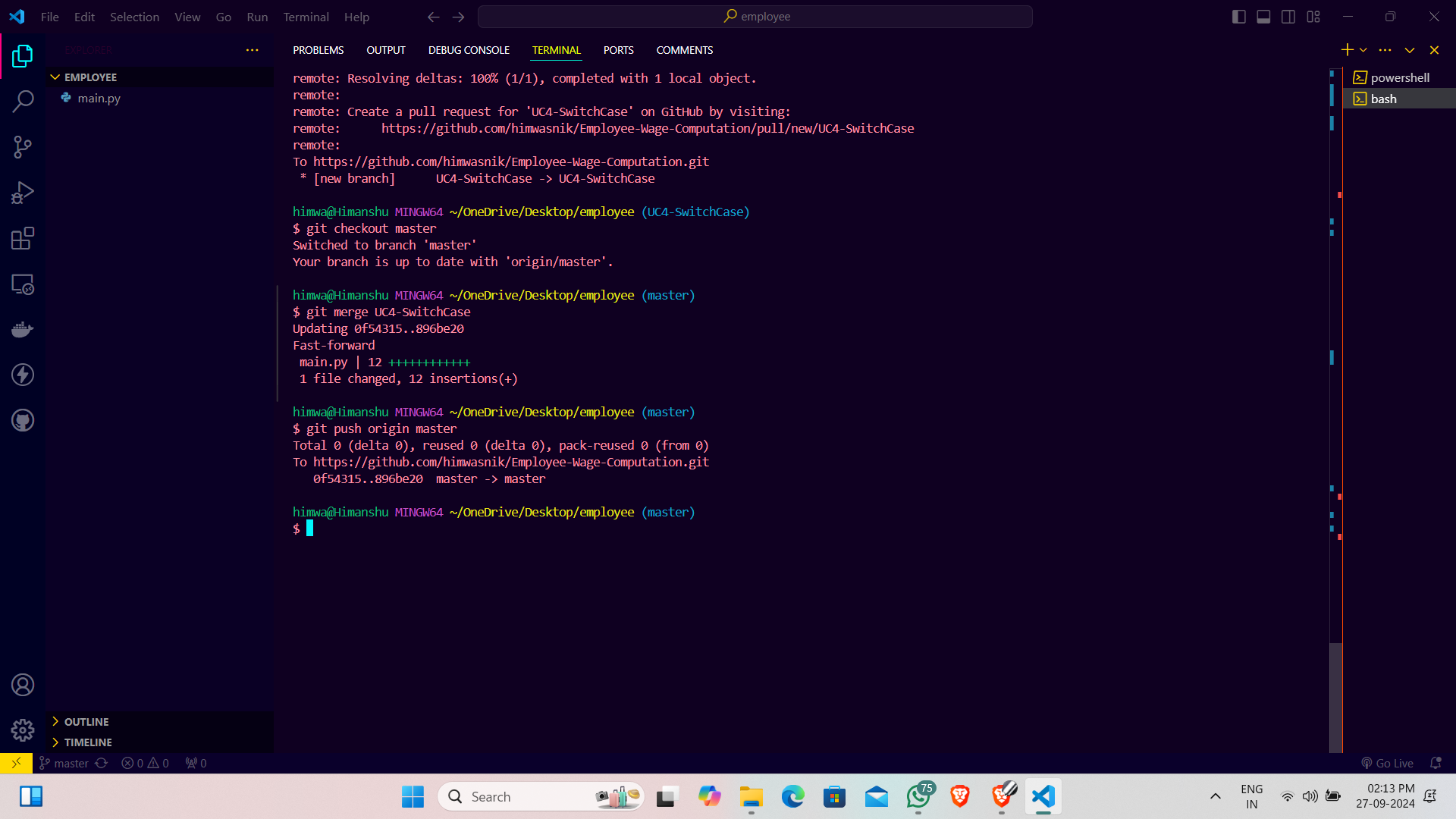
}

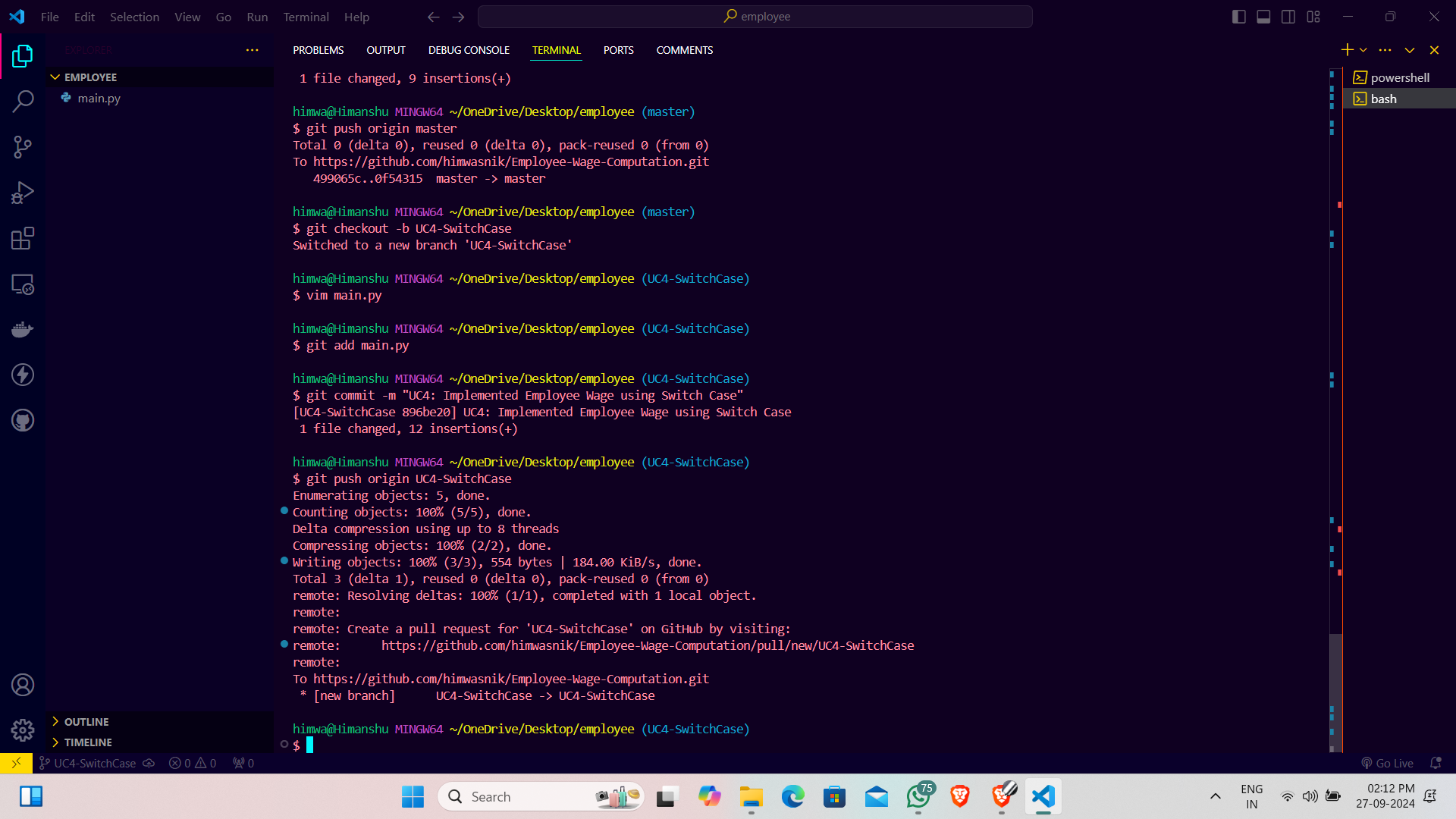
return switcher.get(employee\_type, 0) # Default for absent

employee\_type = random.choice([0, 1, 2]) # 0: Absent, 1: Full-time, 2: Part-time

print(f"Employee Wage (Switch Case): {employee\_wage(employee\_type)}")

**Explanation:-**

Monthly Wage Calculation (UC 4): monthly\_wage() calculates the wage for 20 working days in a month.



***UC5: Calculate Wages for a Month***

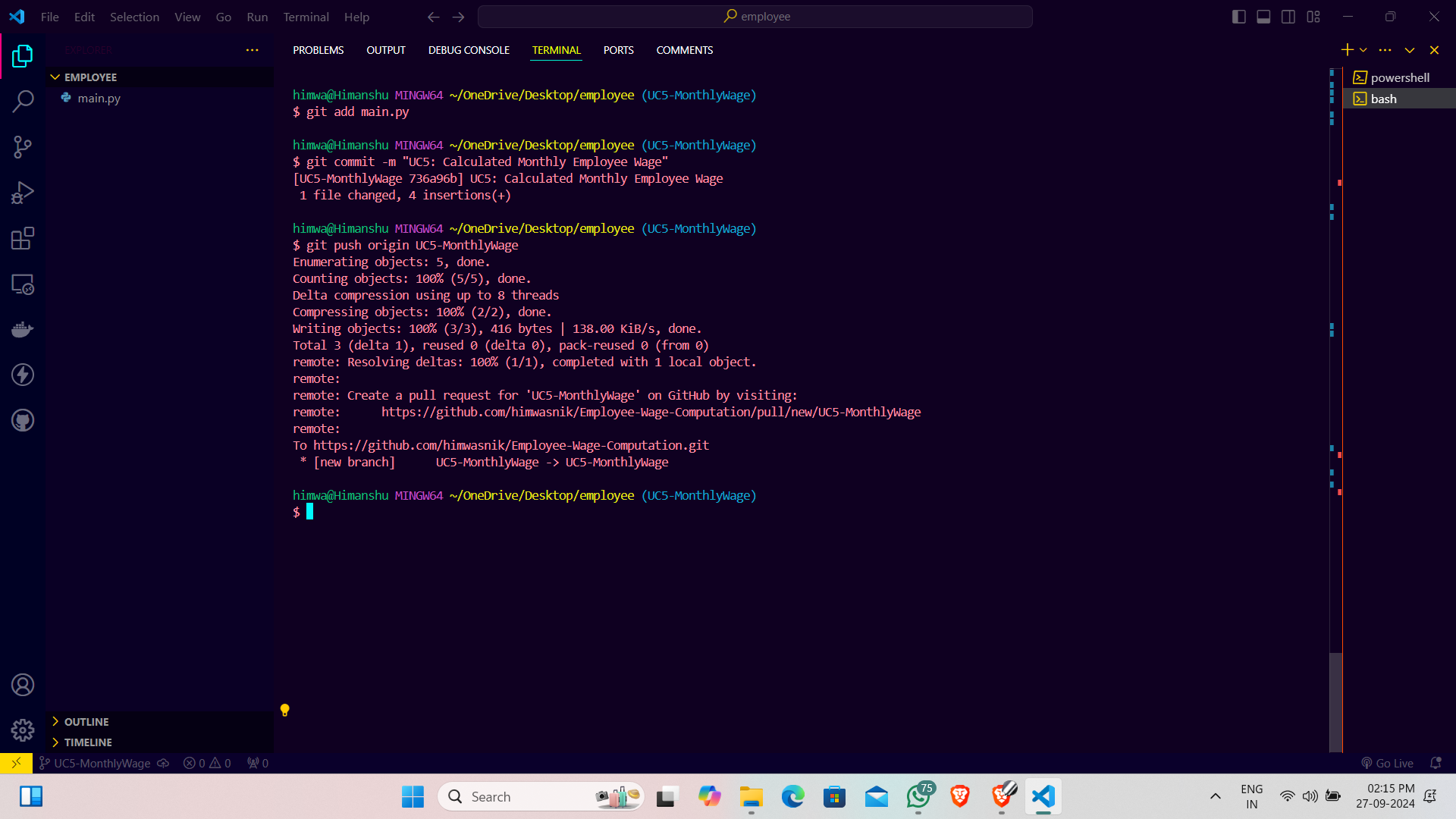
working\_days\_per\_month = 20

monthly\_wage = working\_days\_per\_month \* daily\_wage

print(f"Monthly Employee Wage: {monthly\_wage}")

**Explanation:-**

**Conditional Monthly Wage Calculation (UC 5)**: conditional\_monthly\_wage() keeps track of both total days and hours worked, stopping when either condition exceeds the monthly limits.



***UC6: Calculate Wages Till Condition***

max\_hours\_per\_month = 100

total\_working\_days = 0

total\_working\_hours = 0

while total\_working\_days < 20 and total\_working\_hours < max\_hours\_per\_month:

hours\_worked = random.choice([0, 4, 8]) # 0: Absent, 4: Part-time, 8: Full-time

total\_working\_hours += hours\_worked

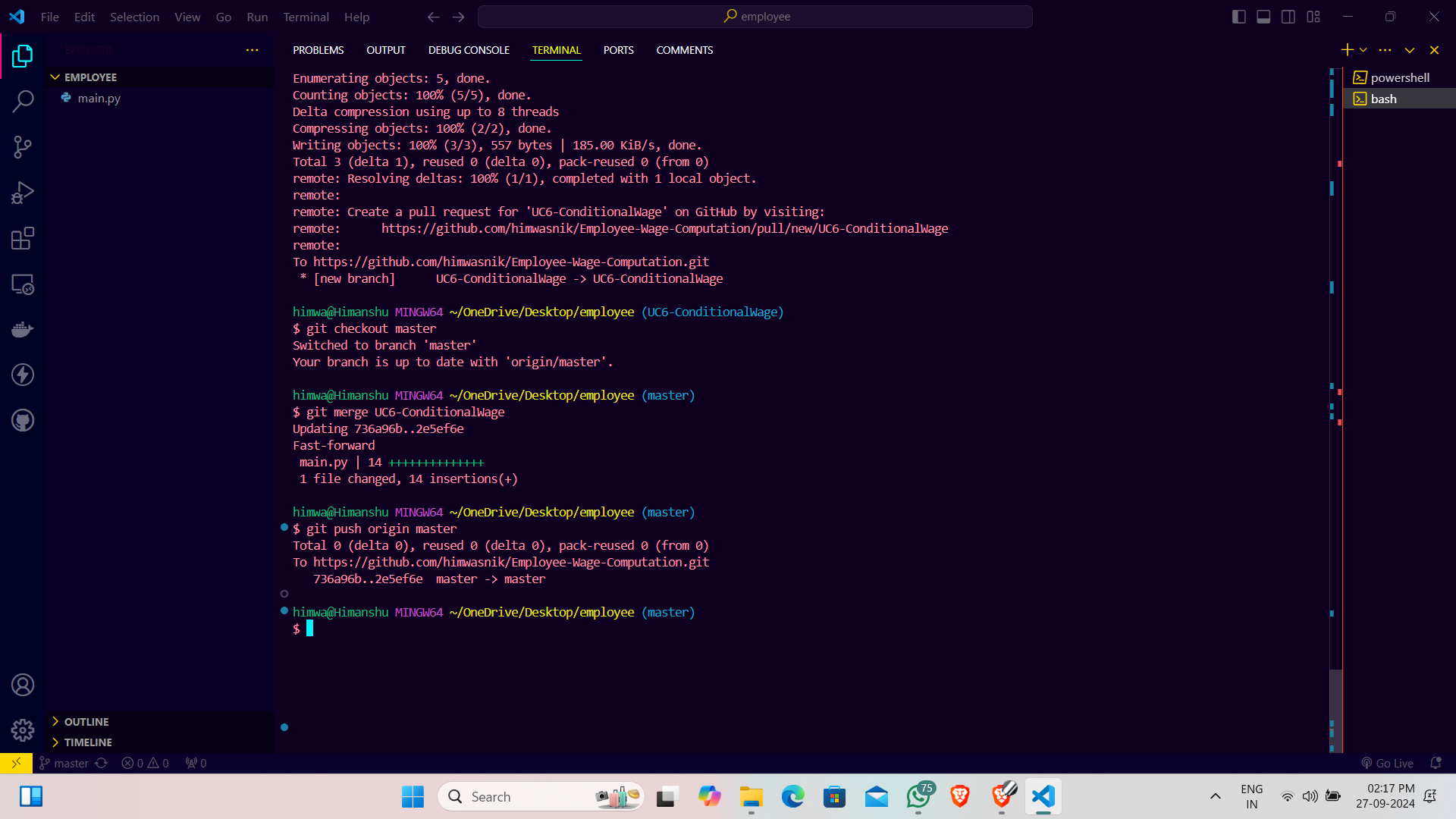
total\_working\_days += 1

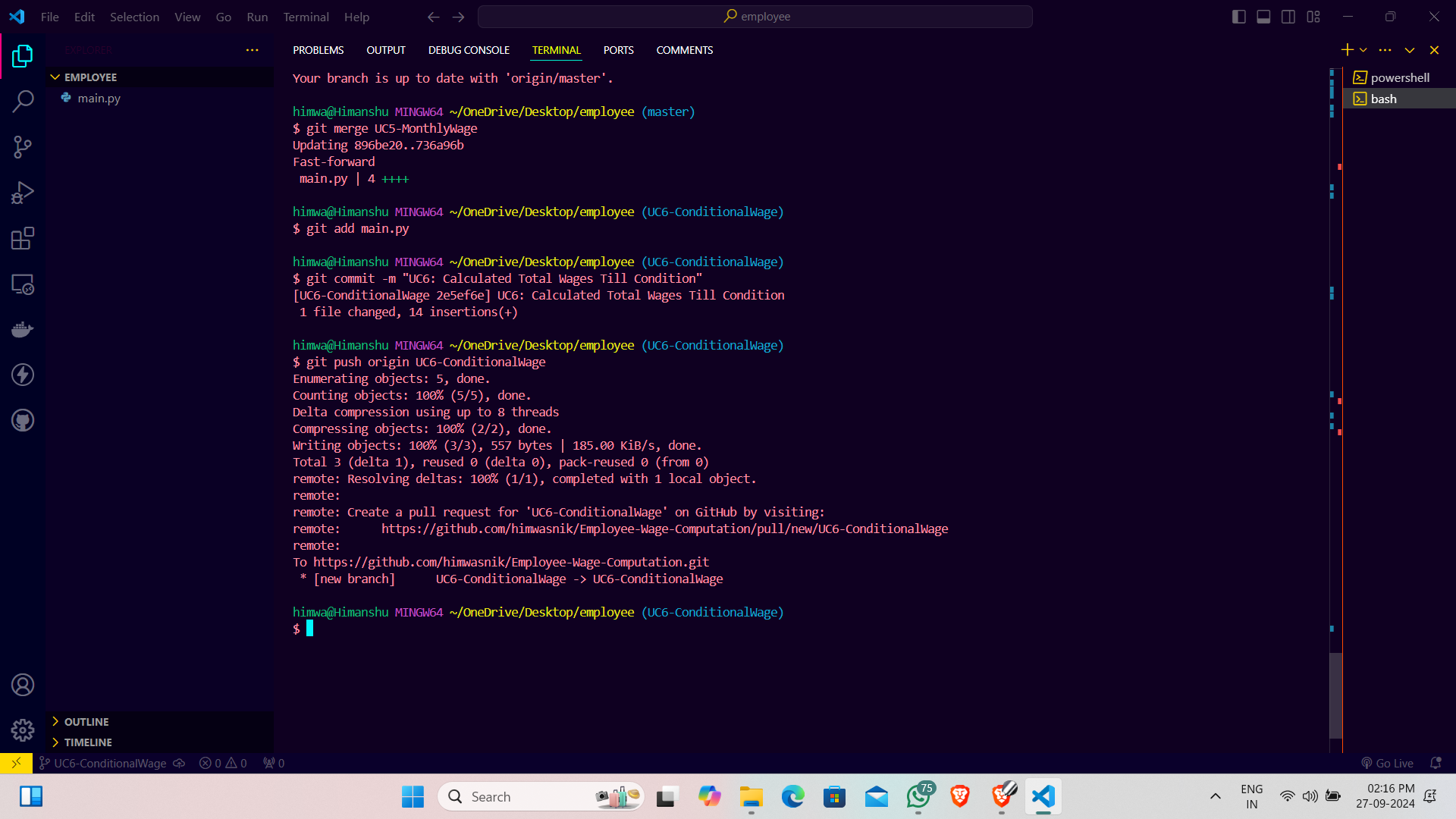
total\_wage = total\_working\_hours \* wage\_per\_hour

print(f"Total Employee Wage for the Month: {total\_wage}")

**Explanation:-**

**Main Method (UC 6)**: compute\_employee\_wage() is the main entry point, calling conditional\_monthly\_wage() and printing the total monthly wage based on the conditions.





***UC7: Refactor code using class method***

class EmployeeWageComputation:

wage\_per\_hour = 20

full\_day\_hour = 8

part\_time\_hour = 4

@classmethod

def calculate\_wage(cls, employee\_type):

if employee\_type == 1: # Full-time

return cls.full\_day\_hour \* cls.wage\_per\_hour

elif employee\_type == 2: # Part-time

return cls.part\_time\_hour \* cls.wage\_per\_hour

else:

return 0 # Absent

employee\_type = random.choice([0, 1, 2]) # 0: Absent, 1: Full-time, 2: Part-time

wage = EmployeeWageComputation.calculate\_wage(employee\_type)

print(f"Employee Wage (Class Method): {wage}")

