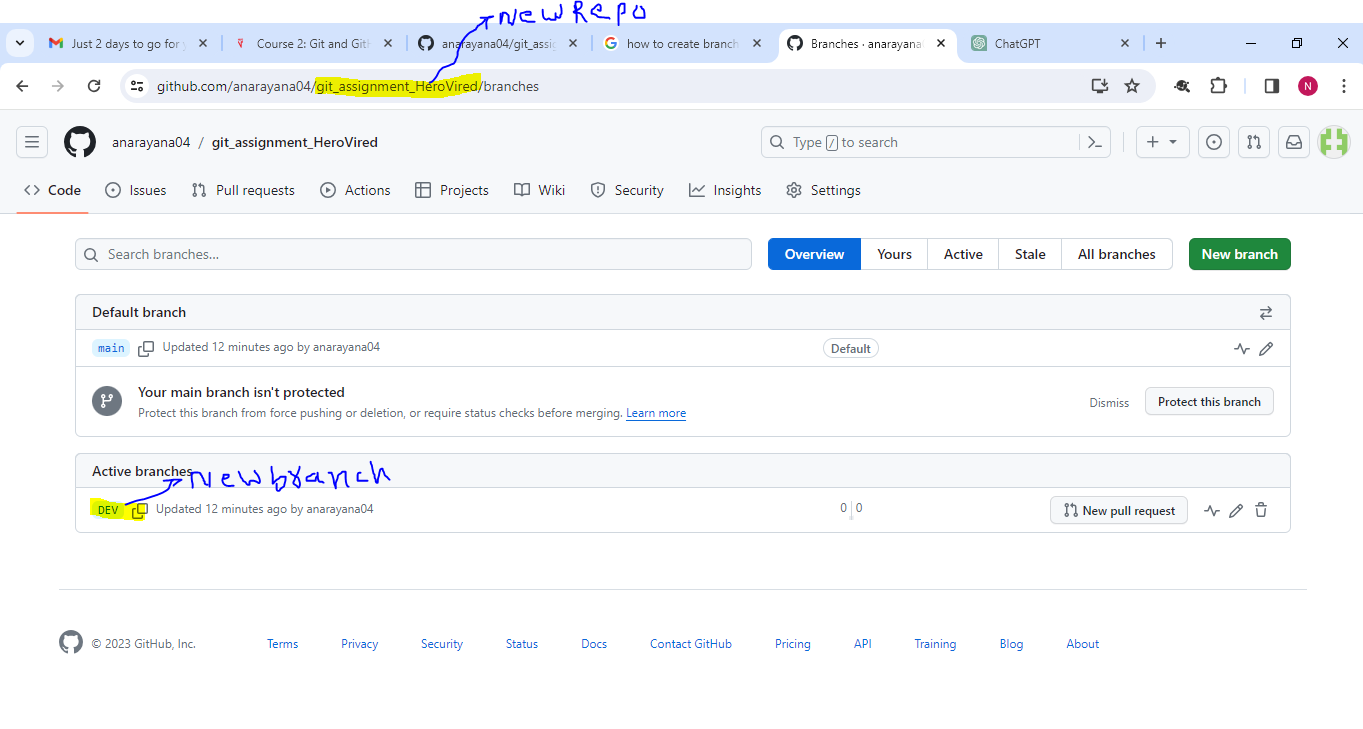
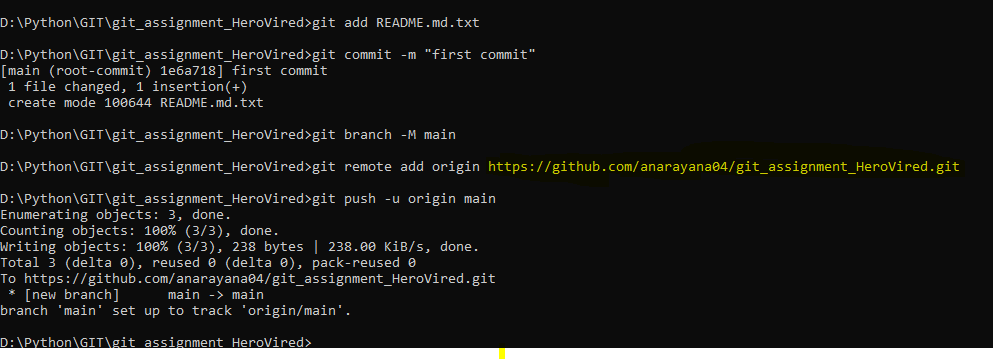
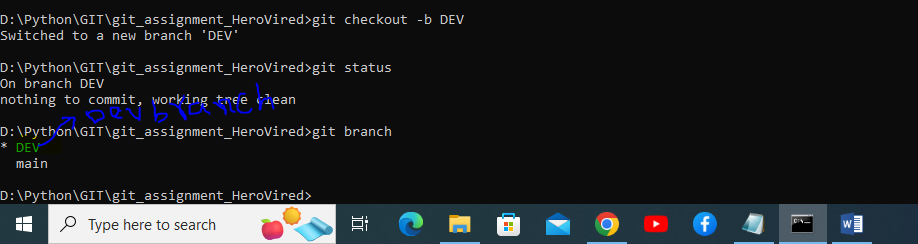
git\_assignment\_HeroVired:

Git link: <https://github.com/anarayana04/git_assignment_HeroVired>

**Q.1:**You are part of a development team working on a Python application called "CalculatorPlus." The application provides basic arithmetic operations, such as addition, subtraction, multiplication, and division. Your task is to implement a new feature that adds support for calculating the square root of a number.







CalulatorPlus program:

def add(x, y):

    return x + y

def subtract(x, y):

    return x - y

def multiply(x, y):

    return x \* y

def divide(x, y):

    if y != 0:

        return x / y

    else:

        return "Cannot divide by zero"

def calculator\_plus():

    print("Welcome to CalculatorPlus!")

    while True:

        print("\nSelect operation:")

        print("1. Addition")

        print("2. Subtraction")

        print("3. Multiplication")

        print("4. Division")

        print("5. Exit")

        choice = input("Enter choice (1/2/3/4/5): ")

        if choice == '5':

            print("Exiting CalculatorPlus. Goodbye!")

            break

        if choice in ('1', '2', '3', '4'):

            try:

                num1 = float(input("Enter first number: "))

                num2 = float(input("Enter second number: "))

            except ValueError:

                print("Invalid input. Please enter valid numbers.")

                continue

            if choice == '1':

                print(f"{num1} + {num2} = {add(num1, num2)}")

            elif choice == '2':

                print(f"{num1} - {num2} = {subtract(num1, num2)}")

            elif choice == '3':

                print(f"{num1} \* {num2} = {multiply(num1, num2)}")

            elif choice == '4':

                result = divide(num1, num2)

                print(f"{num1} / {num2} = {result}")

            else:

                print("Invalid input. Please enter a valid choice.")

        else:

            print("Invalid input. Please enter a valid choice.")

if \_\_name\_\_ == "\_\_main\_\_":

    calculator\_plus()

Output:

Enter choice (1/2/3/4/5): 1

Enter first number: 1

Enter second number: 1

1.0 + 1.0 = 2.0

Select operation:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter choice (1/2/3/4/5):

Enter choice (1/2/3/4/5): 2

Enter first number: 2

Enter second number: 2

2.0 - 2.0 = 0.0

Select operation:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter choice (1/2/3/4/5):

Enter choice (1/2/3/4/5): 3

Enter first number: 2

Enter second number: 2

2.0 \* 2.0 = 4.0

Select operation:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter choice (1/2/3/4/5):

Enter choice (1/2/3/4/5): 4

Enter first number: 2

Enter second number: 2

2.0 / 2.0 = 1.0

Select operation:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter choice (1/2/3/4/5):

Enter choice (1/2/3/4/5): 4

Enter first number: 2

Enter second number: 2

2.0 / 2.0 = 1.0

Select operation:

1. Addition

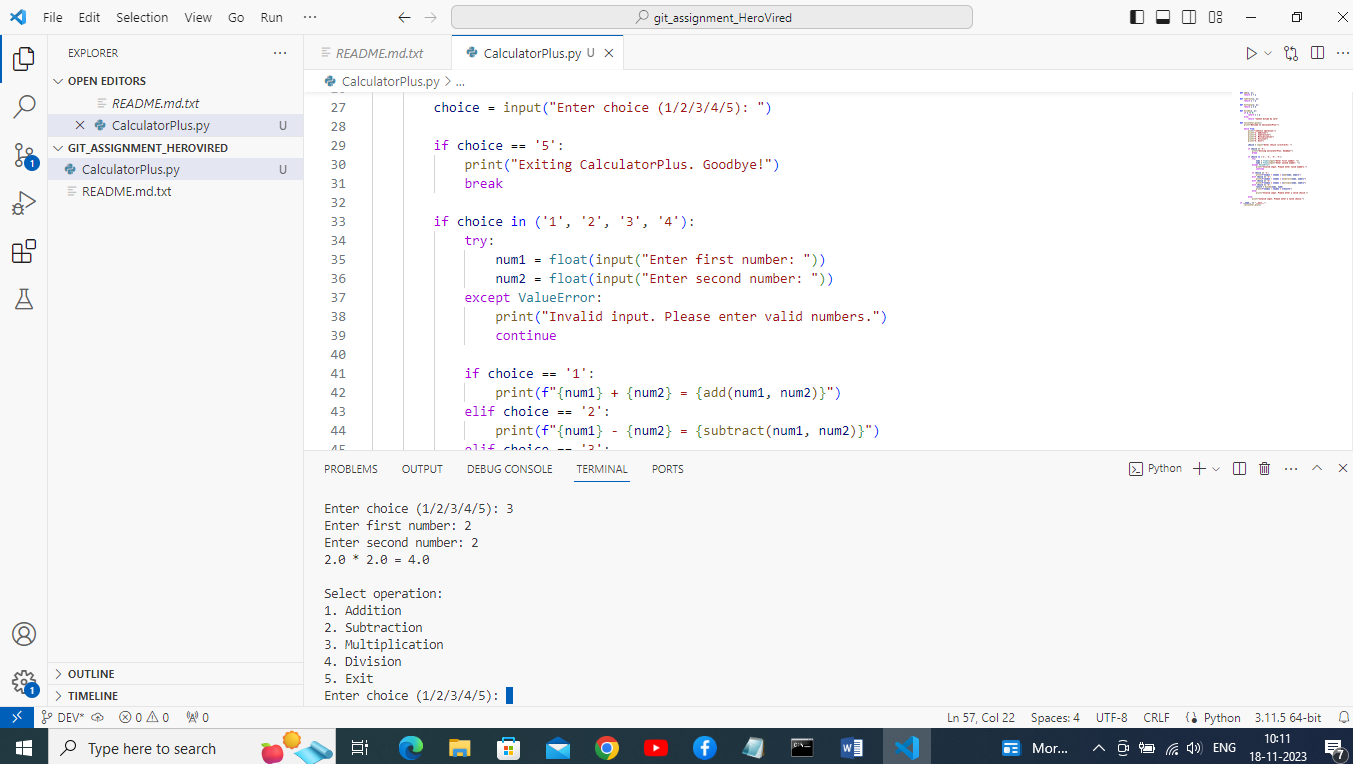
2. Subtraction

3. Multiplication

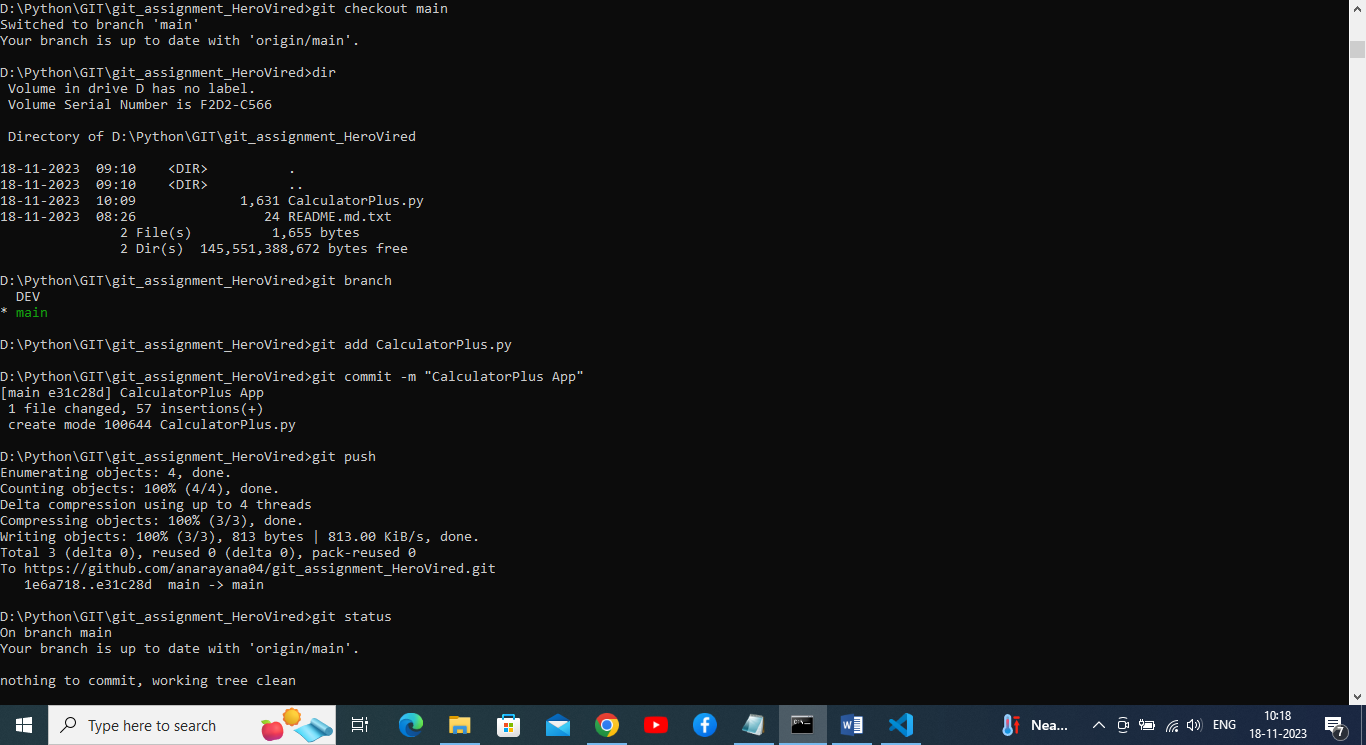
4. Division

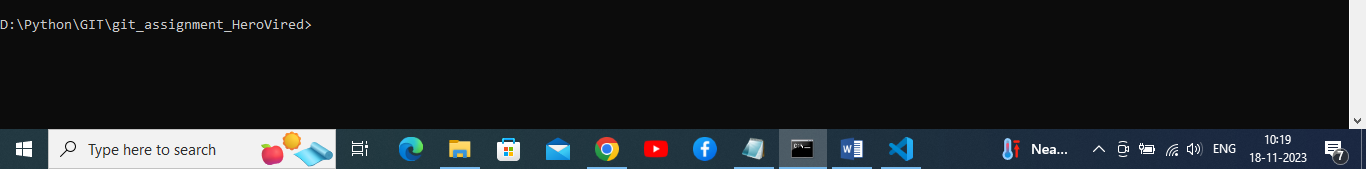
5. Exit

Enter choice (1/2/3/4/5):



Pushed code to Git hub





Square root code added in CalculatorPlus.py program file.

import math

def add(x, y):

    return x + y

def subtract(x, y):

    return x - y

def multiply(x, y):

    return x \* y

def divide(x, y):

    if y != 0:

        return x / y

    else:

        return "Error: Cannot divide by zero"

def square\_root(x):

    if x >= 0:

        return math.sqrt(x)

    else:

        return "Error: Cannot calculate square root of a negative number"

def calculator\_plus():

    print("Welcome to CalculatorPlus!")

    while True:

        print("\nChoose an operation:")

        print("1. Addition")

        print("2. Subtraction")

        print("3. Multiplication")

        print("4. Division")

        print("5. Square Root")

        print("6. Exit")

        choice = input("Enter the operation number (1-6): ")

        if choice == '6':

            print("Exiting CalculatorPlus. Goodbye!")

            break

        if choice in ('1', '2', '3', '4', '5'):

            if choice == '5':

                num = float(input("Enter a number: "))

                result = square\_root(num)

                print(f"Result: {result}")

            else:

                num1 = float(input("Enter the first number: "))

                num2 = float(input("Enter the second number: "))

                if choice == '1':

                    result = add(num1, num2)

                elif choice == '2':

                    result = subtract(num1, num2)

                elif choice == '3':

                    result = multiply(num1, num2)

                elif choice == '4':

                    result = divide(num1, num2)

                print(f"Result: {result}")

        else:

            print("Invalid choice. Please enter a number between 1 and 6.")

if \_\_name\_\_ == "\_\_main\_\_":

    calculator\_plus()

Output:

Enter the operation number (1-6): 5

Enter a number: 8

Result: 2.8284271247461903

Choose an operation:

1. Addition

2. Subtraction

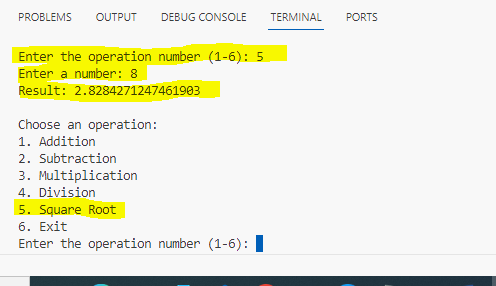
3. Multiplication

4. Division

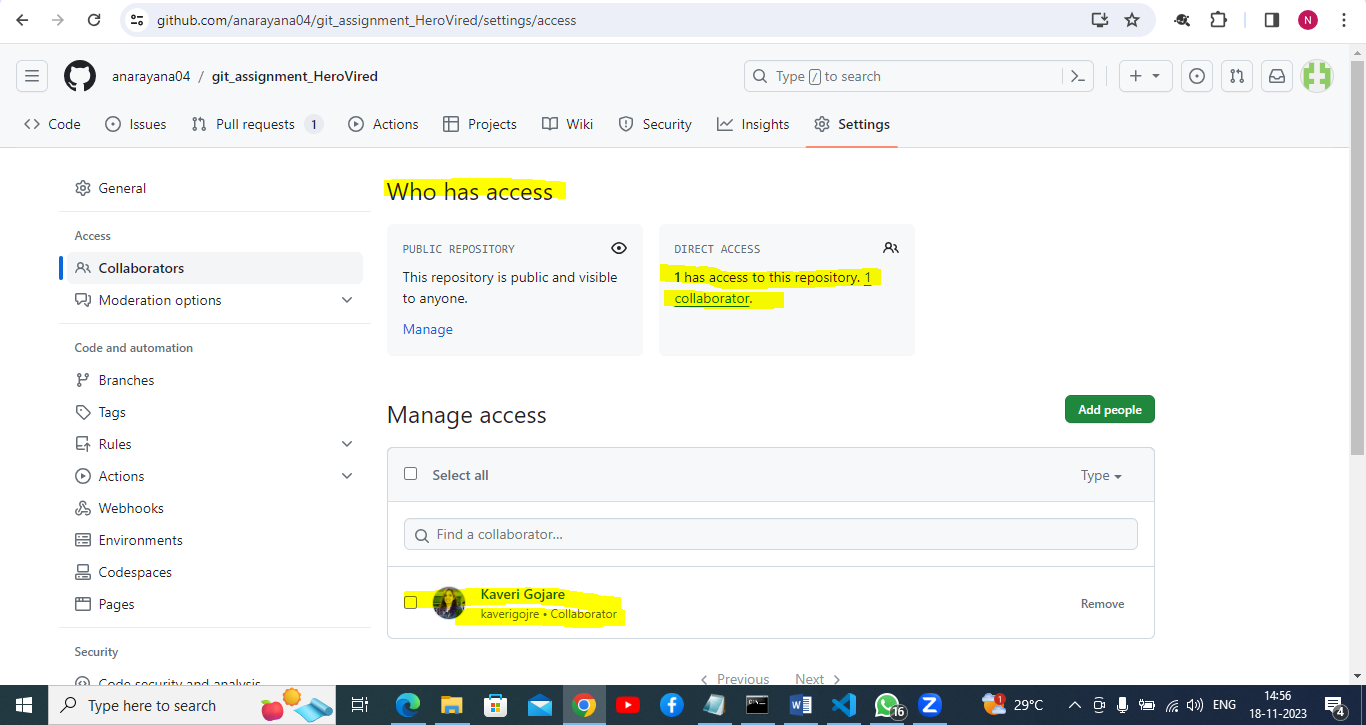
5. Square Root

6. Exit

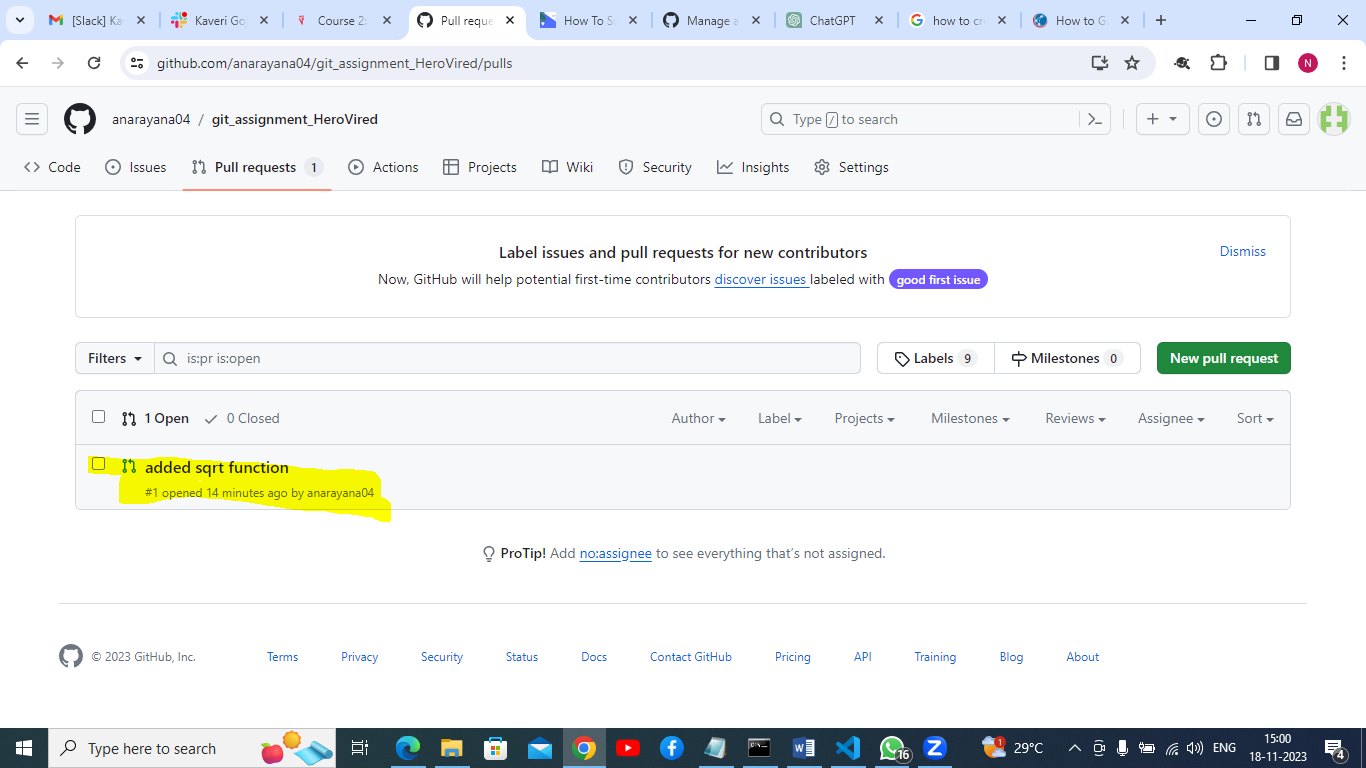
Enter the operation number (1-6):



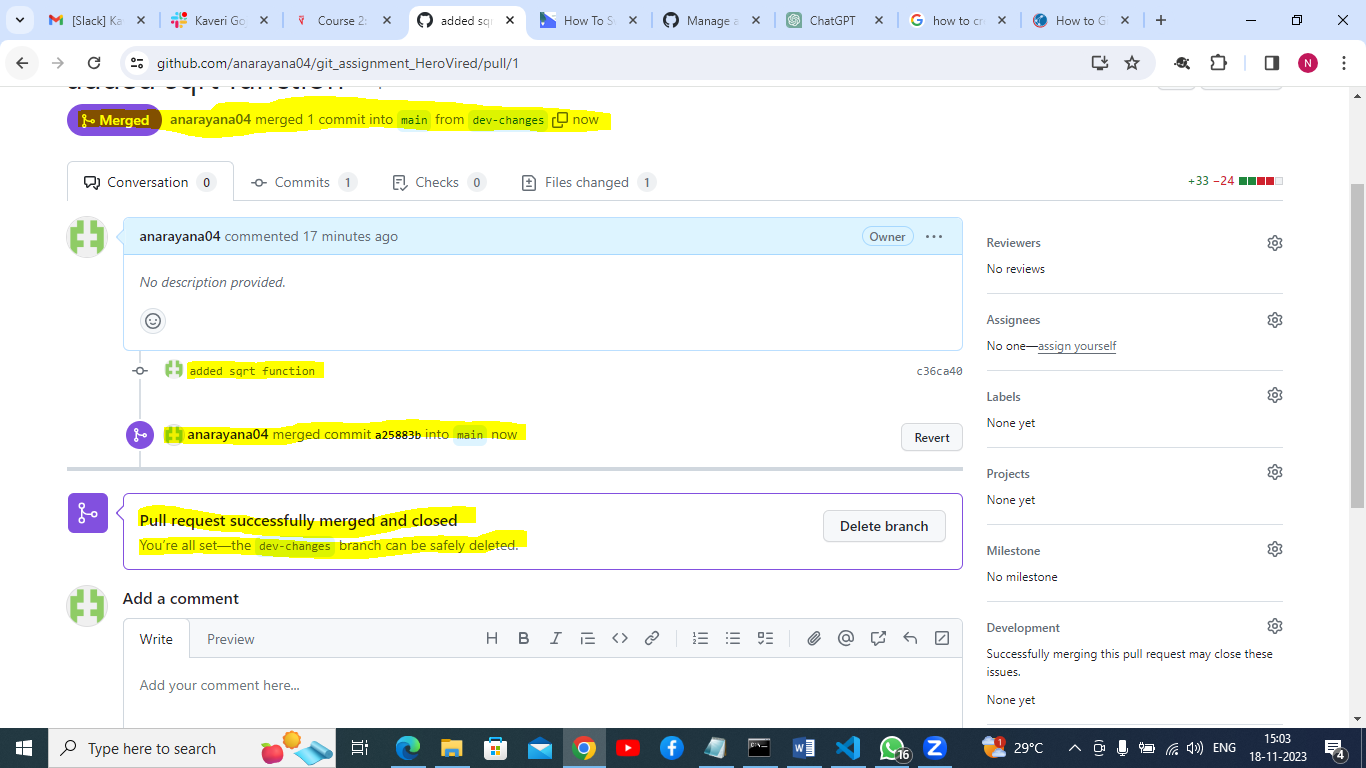
Added team member for review added code.



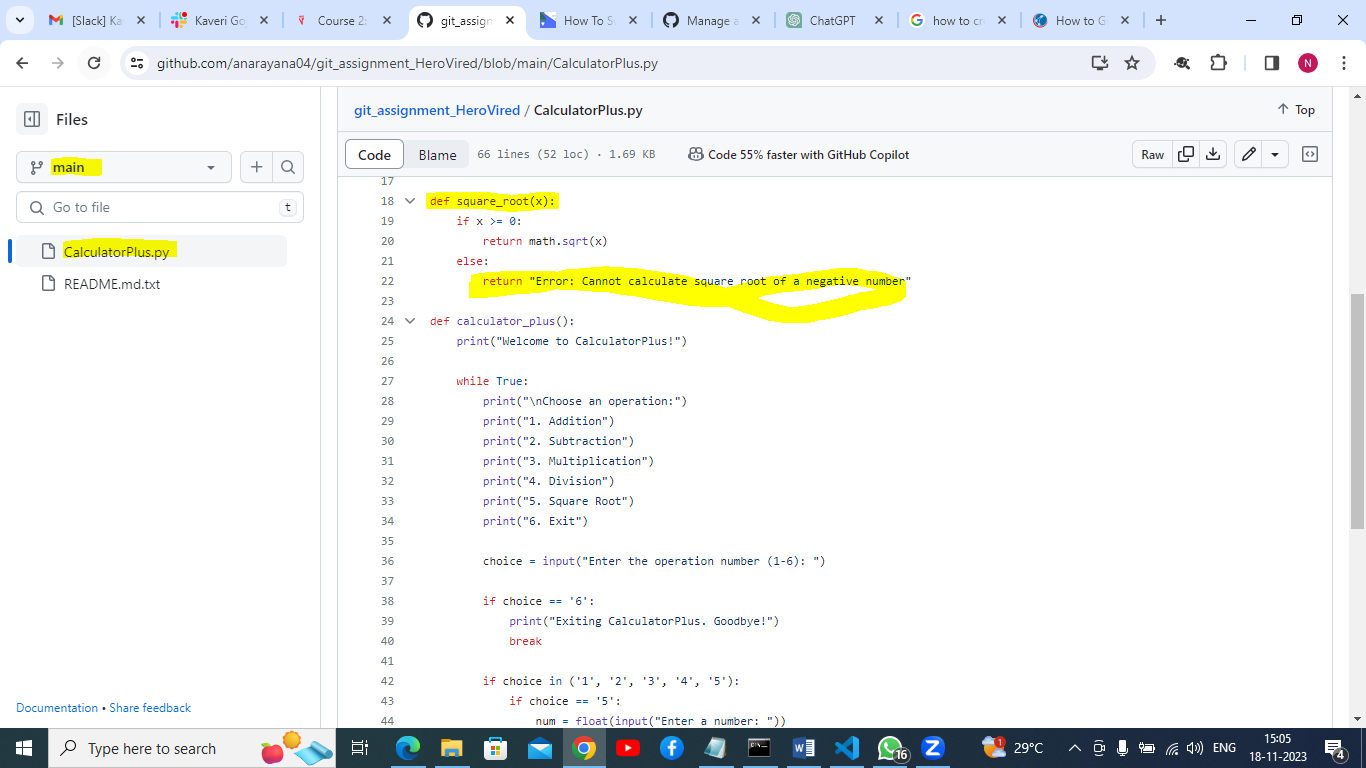
Crated pull request to merge from dev-changes to main repo.



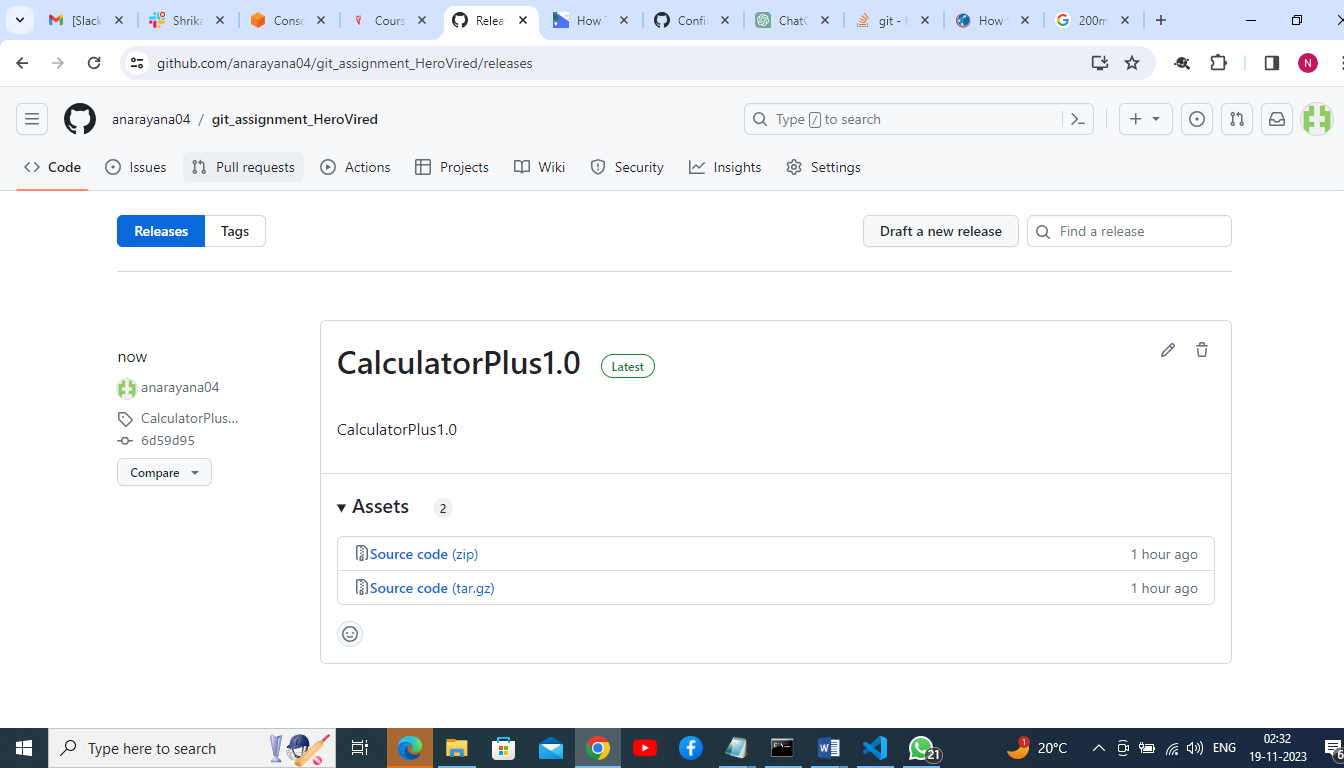
Merged successfully.



From main branch added code.



Release created:

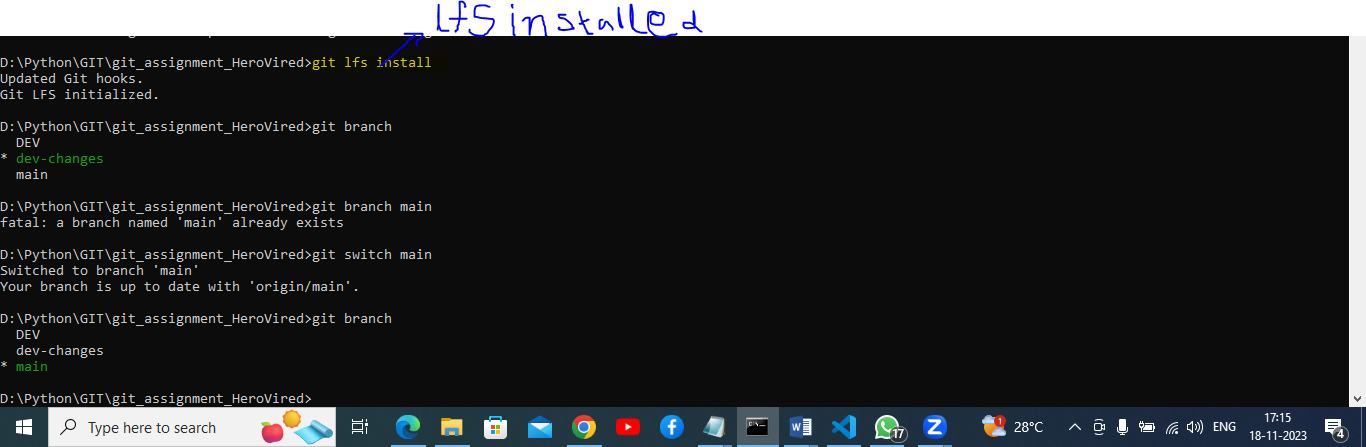


**Q.2:**For a project that deals with large binary files, integrate Git LFS (Large File Storage) to handle these files efficiently. Demonstrate how to add, commit, and push binary files to the repository, ensuring they are tracked by Git LFS correctly. Clone the repository on another machine to verify that the binary files are downloaded correctly.

Git repo link: https://github.com/anarayana04/git\_assignment\_HeroVired/commit/e4d2e531423d6794140ffe3cd97fc2008b0f6796

### Initialize Git LFS in your repository

git lfs install

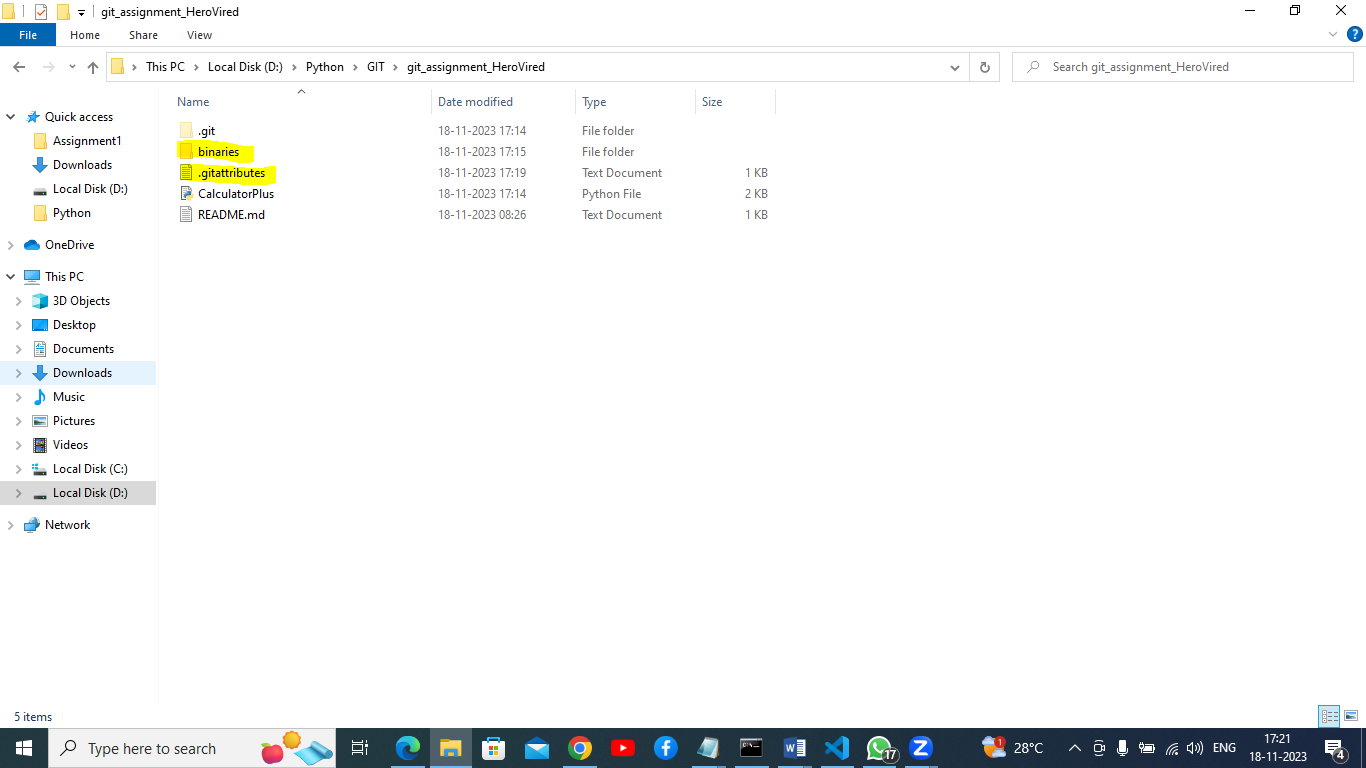


**Track binary files with Git LFS**

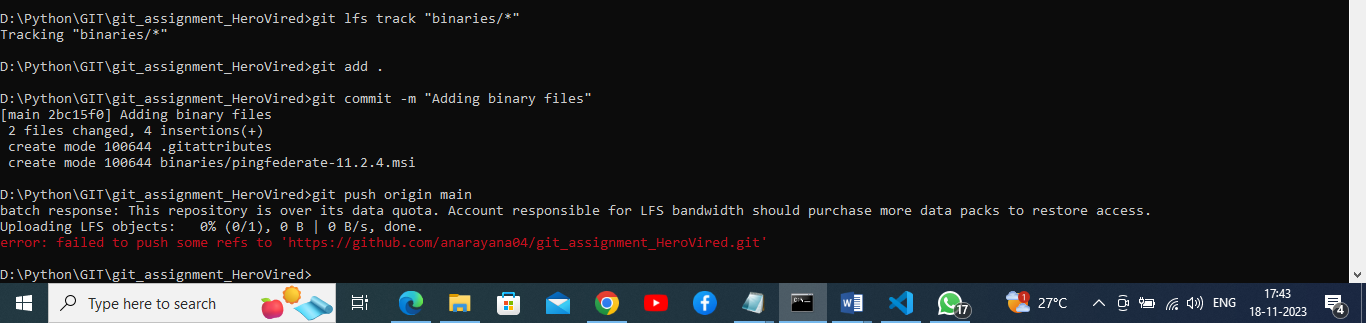
git lfs track "binaries/\*"



This command creates or updates the **.gitattributes** file to indicate which files should be tracked using Git LFS.

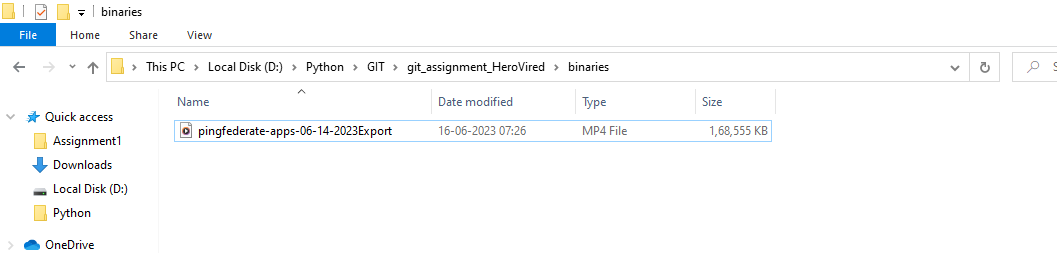


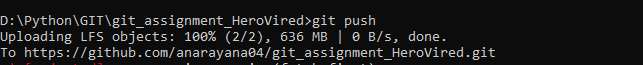
### Add, commit, and push binary files

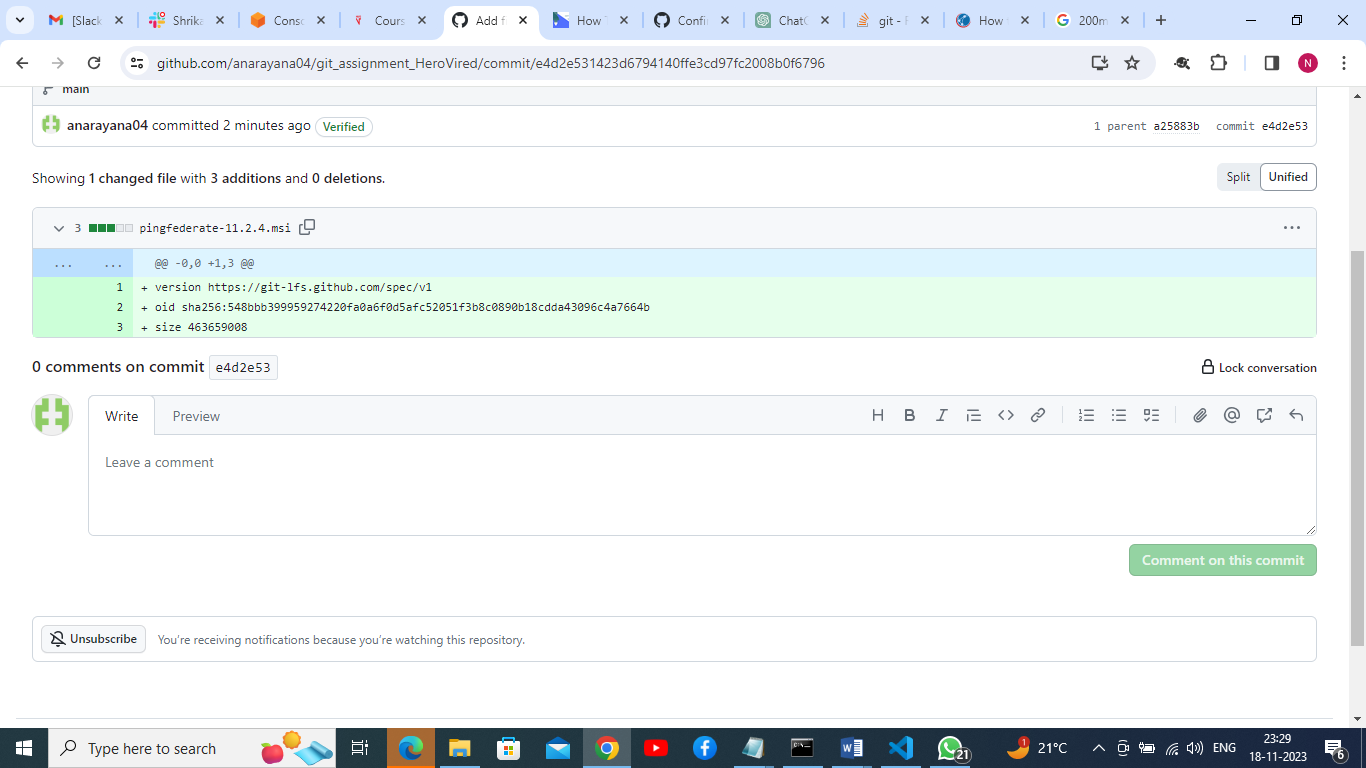


Getting error due to trying to more than 400MB file

I am going to delete that much space file and adding 164MB file.





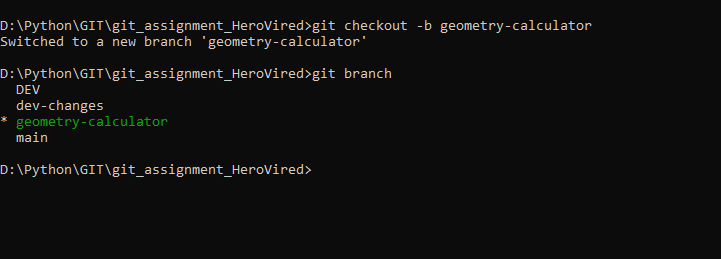


Due to space issue deleted large binary files from repo and local.

**Q.3:**In this same GitHub repository, create a new branch ‘geometry-calculator’, we'll work on a simple Python program that calculates the area of a circle and the area of a rectangle. We'll use Git stash to switch between working on multiple features (calculating circle area and calculating rectangle area) without committing incomplete changes.

Git url link: https://github.com/anarayana04/git\_assignment\_HeroVired/tree/geometry-calculator

Created geometry-calculator branch



Code :

import math

class GeometryCalculator:

    def calculate\_circle\_area(self, radius):

        return math.pi \* radius \*\* 2

    def calculate\_rectangle\_area(self, length, width):

        return length \* width

if \_\_name\_\_ == "\_\_main\_\_":

    calculator = GeometryCalculator()

    # Calculate the area of a circle

    radius = 5

    circle\_area = calculator.calculate\_circle\_area(radius)

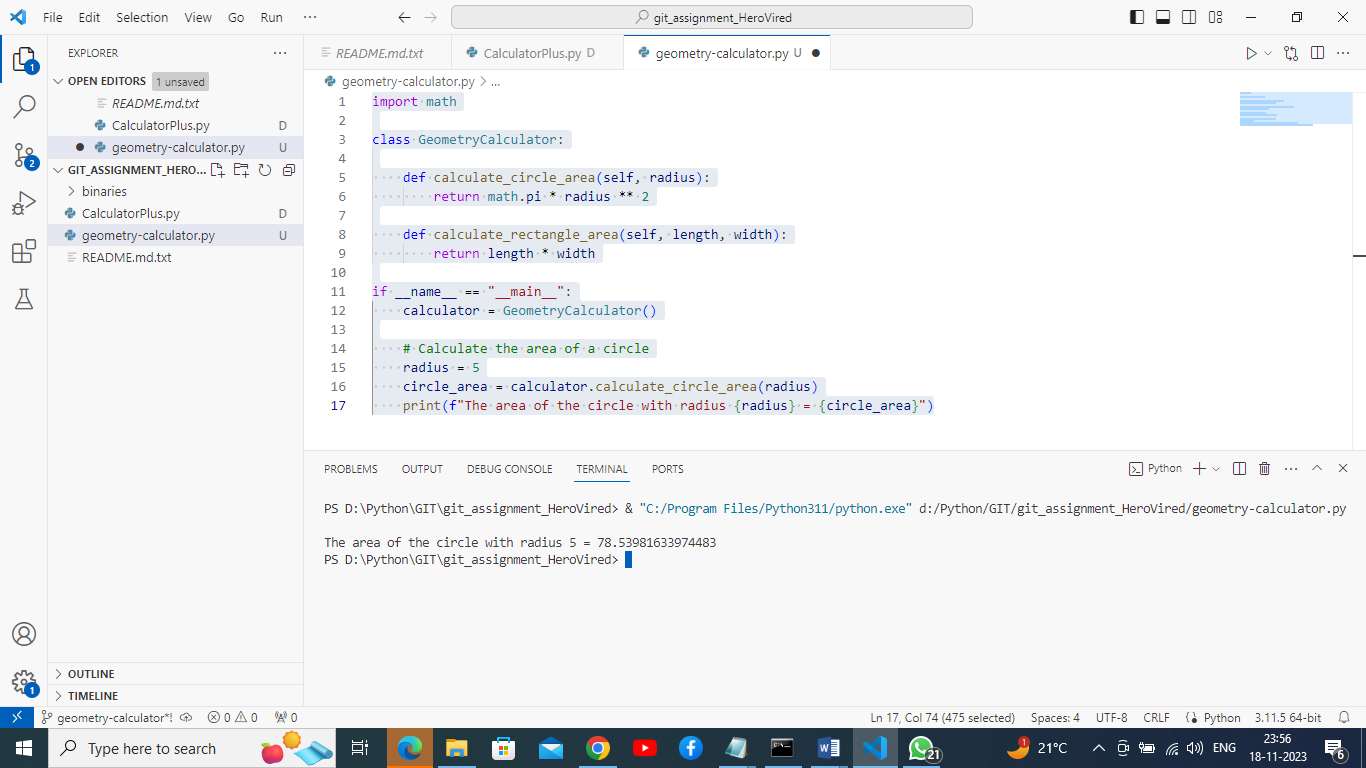
    print(f"The area of the circle with radius {radius} = {circle\_area}")

Output:

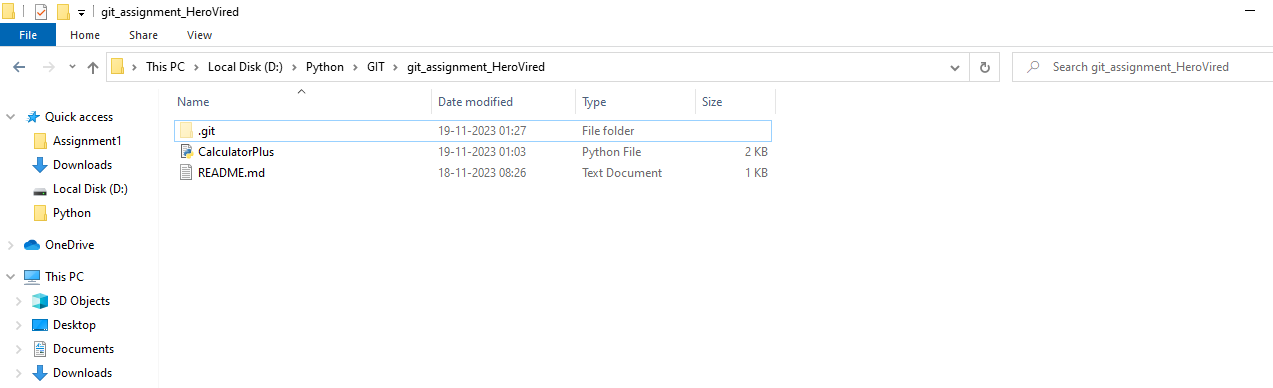
PS D:\Python\GIT\git\_assignment\_HeroVired> & "C:/Program Files/Python311/python.exe" d:/Python/GIT/git\_assignment\_HeroVired/geometry-calculator.py

The area of the circle with radius 5 = 78.53981633974483

PS D:\Python\GIT\git\_assignment\_HeroVired>



Cleaned circle-area.py file from local repo.



Rectangle area code:

def calculate\_rectangle\_area(length, width):

    area = length \* width

    return area

# Given values

length = 10

width = 6

# Calculate and print the area

area = calculate\_rectangle\_area(length, width)

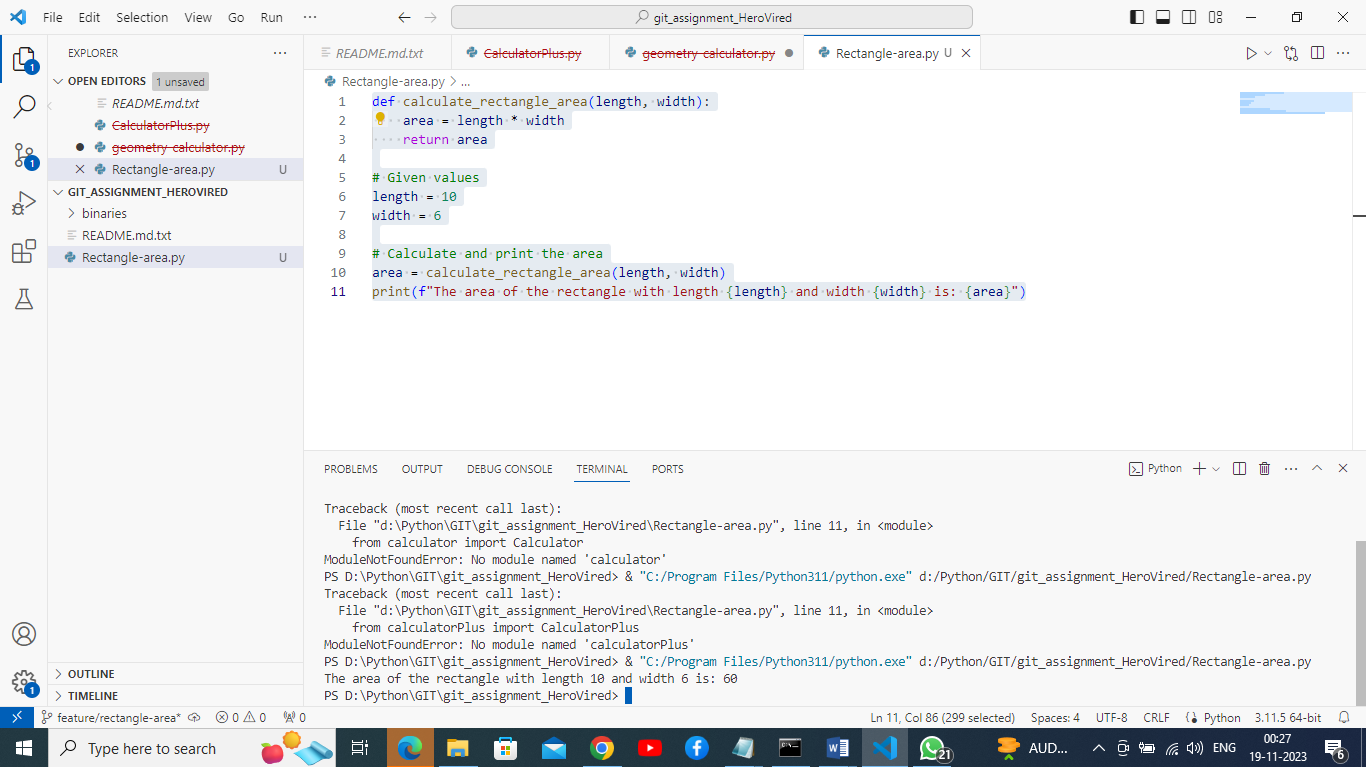
print(f"The area of the rectangle with length {length} and width {width} is: {area}")

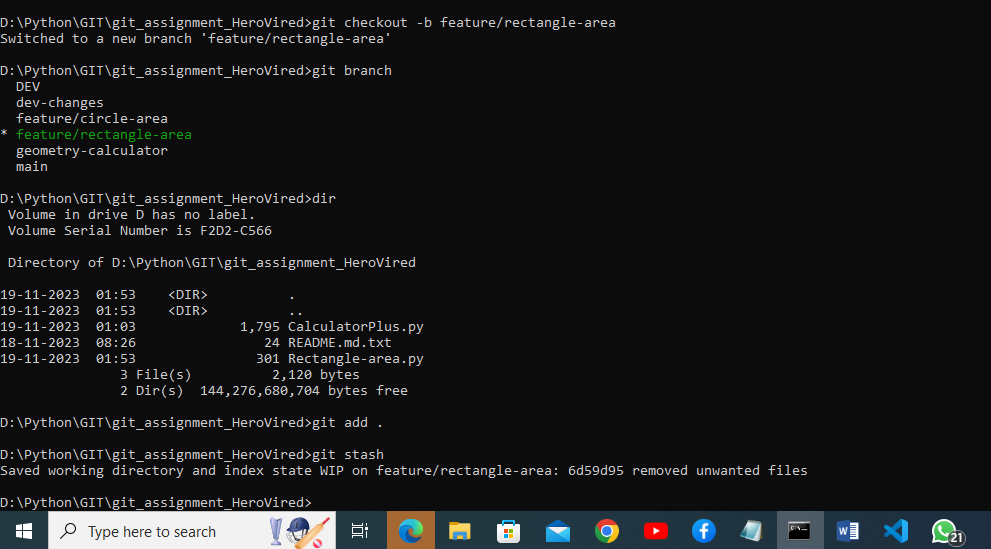
Output:

PS D:\Python\GIT\git\_assignment\_HeroVired> & "C:/Program Files/Python311/python.exe" d:/Python/GIT/git\_assignment\_HeroVired/Rectangle-area.py

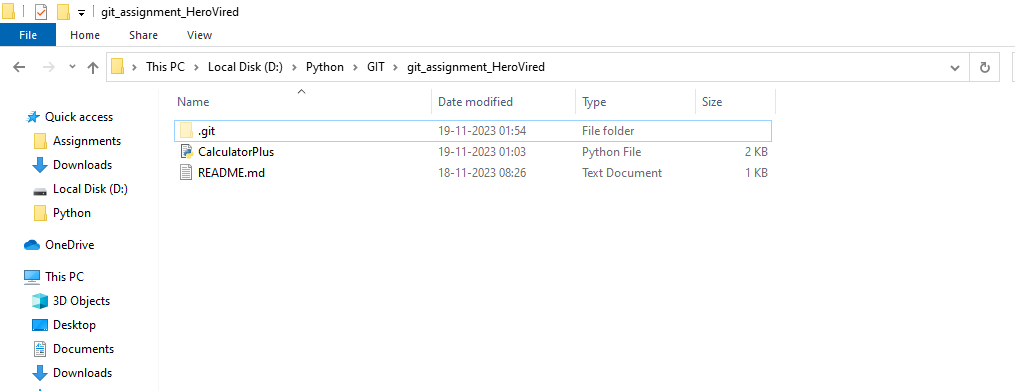
The area of the rectangle with length 10 and width 6 is: 60

PS D:\Python\GIT\git\_assignment\_HeroVired>





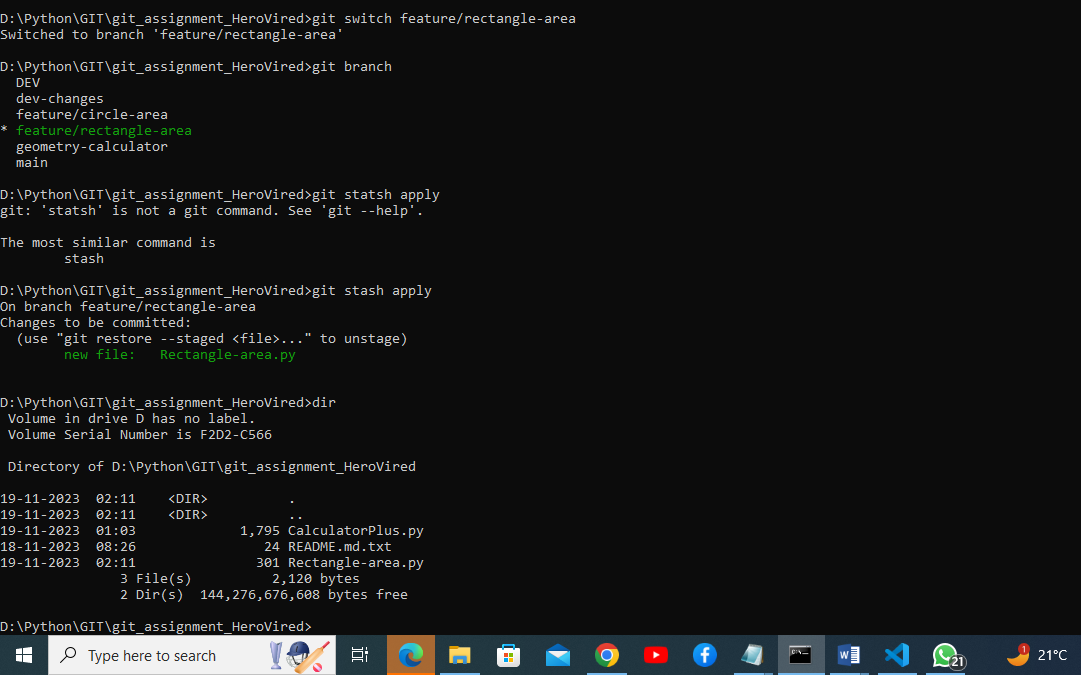
Working directory:

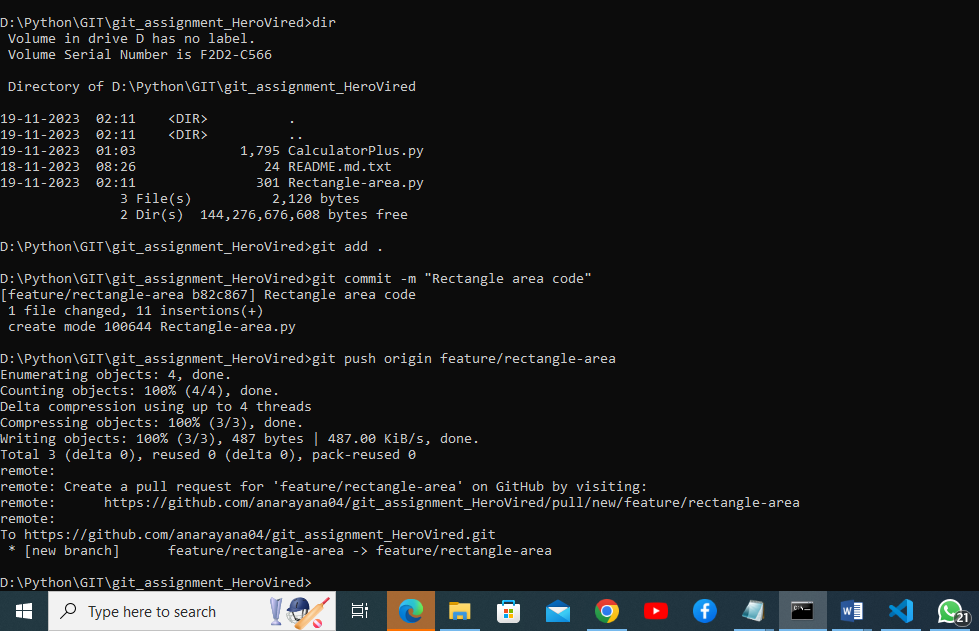


Commit and puch circle area code :

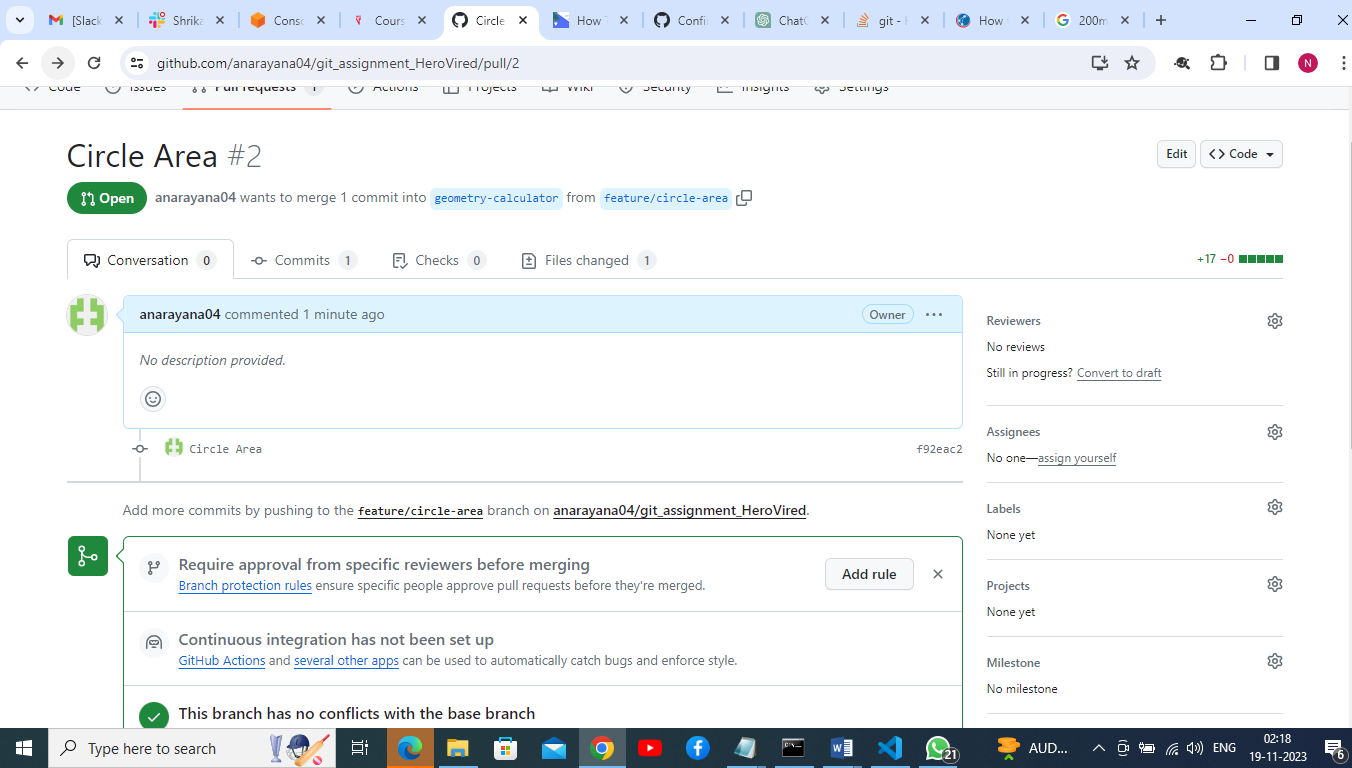


Switching to rectangle area branch

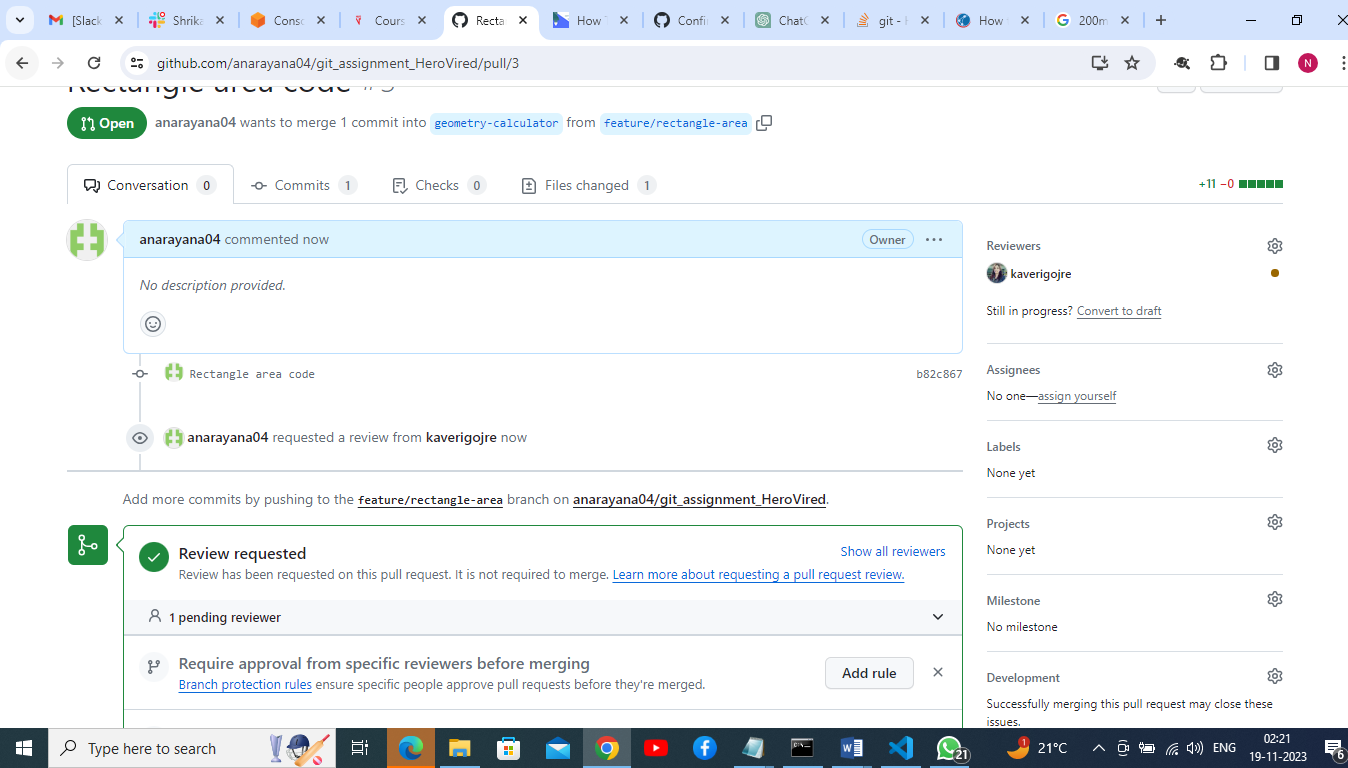




Created pull request for circle area code.



Created pull request for rectangle area.



And assigned team member(Kaveri) for review.

Once reviewed merged changes.

