

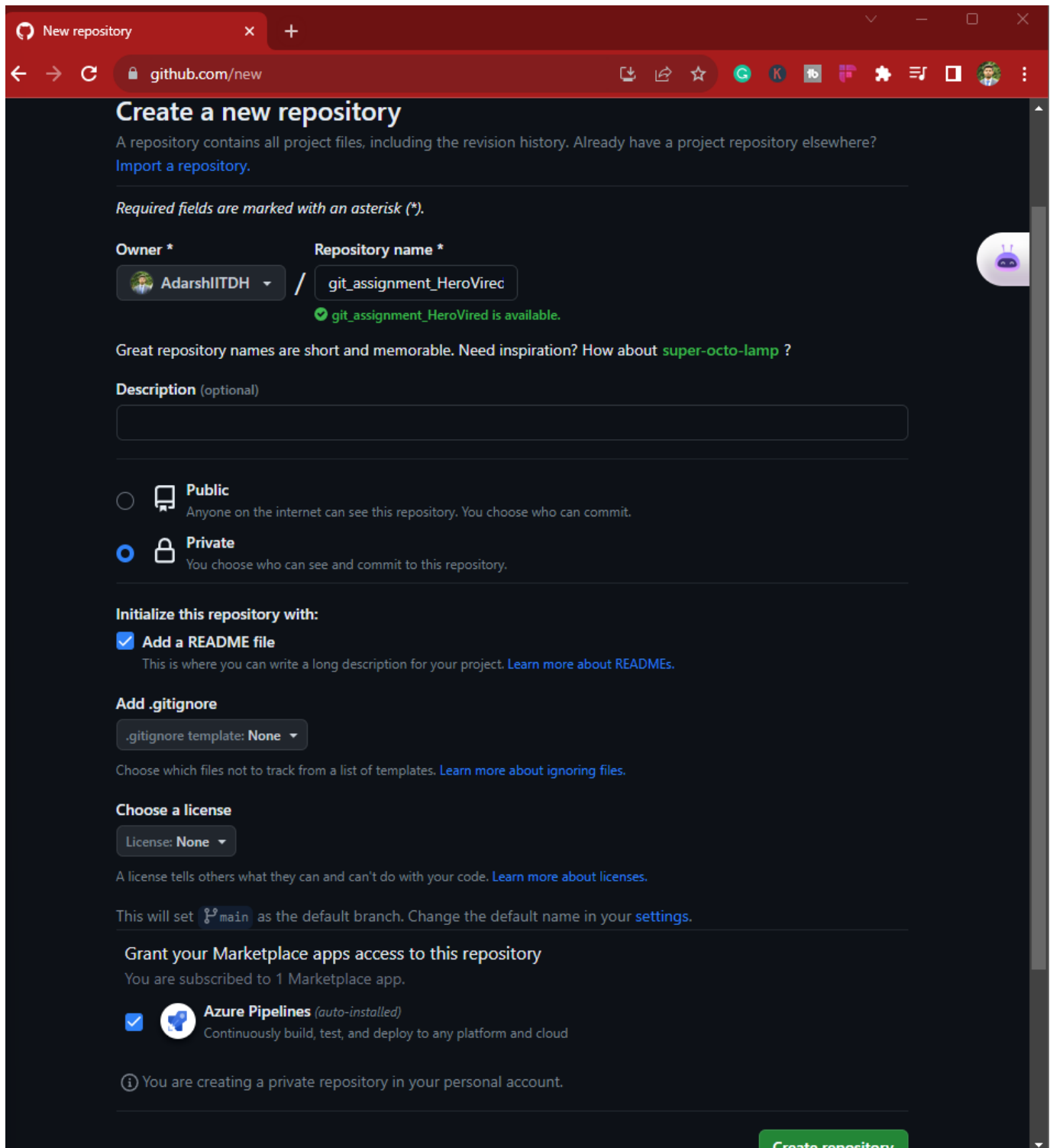
Git and GitHub

https://github.com/AdarshIITDH/git_assignment_HeroVired.git

Question 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.

- a) Create a repository name: git_assignment_HeroVired



New repository

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

AdarshIITDH / git_assignment_HeroVired

git_assignment_HeroVired is available.

Great repository names are short and memorable. Need inspiration? How about [super-octo-lamp](#) ?

Description (optional)

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license


License: None


A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Grant your Marketplace apps access to this repository

You are subscribed to 1 Marketplace app.

☒  **Azure Pipelines** (auto-installed)
Continuously build, test, and deploy to any platform and cloud

 You are creating a private repository in your personal account.

Create repository

b) Create a 'dev' branch and add this code.

```
MINGW64:/f/devops/auto/git_assignment_HeroVired
adars@Adarsh MINGW64 /f/devops/auto
$ git clone git@github.com:AdarshIITDH/git_assignment_HeroVired.git
Cloning into 'git_assignment_HeroVired'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

adars@Adarsh MINGW64 /f/devops/auto
$ cd git_assignment_HeroVired/

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git branch dev

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git checkout dev
Switched to branch 'dev'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ ls
README.md

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ nano CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    CalculatorPlus.py

nothing added to commit but untracked files present (use "git add" to track)

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git add CalculatorPlus.py
warning: in the working copy of 'CalculatorPlus.py', LF will be replaced by CRLF
the next time Git touches it

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git commit -m "step-b committed"
[dev 5cddb56] step-b committed
1 file changed, 39 insertions(+)
create mode 100644 CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git push origin dev
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 657 bytes | 657.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/AdarshIITDH/git_assignment_HeroVired/pull/new/de
v
remote:
To github.com:AdarshIITDH/git_assignment_HeroVired.git
 * [new branch]      dev -> dev

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$
```

```
MINGW64:/f/devops/auto/git_assignment_HeroVired
GNU nano 6.4 CalculatorPlus.py Modified
import math
class Calculator:
    def add(self, a, b): return a + b
    def subtract(self, a, b): return a - b
    def multiply(self, a, b): return a * b
    def divide(self, a, b): return a / b

# TODO: Implement the following function to calculate the square root of a number.

    # def square_root(self, x): return math.sqrt(x)

# You need to uncomment the above function and complete its implementation to add the square root feature.

if __name__ == "__main__": calculator = Calculator()

num1 = 16
num2 = 4

print(f"{num1} + {num2} = {calculator.add(num1, num2)}")
print(f"{num1} - {num2} = {calculator.subtract(num1, num2)}")
print(f"{num1} * {num2} = {calculator.multiply(num1, num2)}")
print(f"{num1} / {num2} = {calculator.divide(num1, num2)}")

# TODO: Uncomment and test the square root feature.
# num3 = 25
# print(f"The square root of {num3} = {calculator.square_root(num3)}")
```

- c) Merge this branch with the main branch and make a release of version 1 of the 'calculator plus app'.

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git merge dev
Updating 8bd302b..72f6885
Fast-forward
 CalculatorPlus.py | 28 +++++
 1 file changed, 28 insertions(+)
 create mode 100644 CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git commit -m "step c committing for version-1 release"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

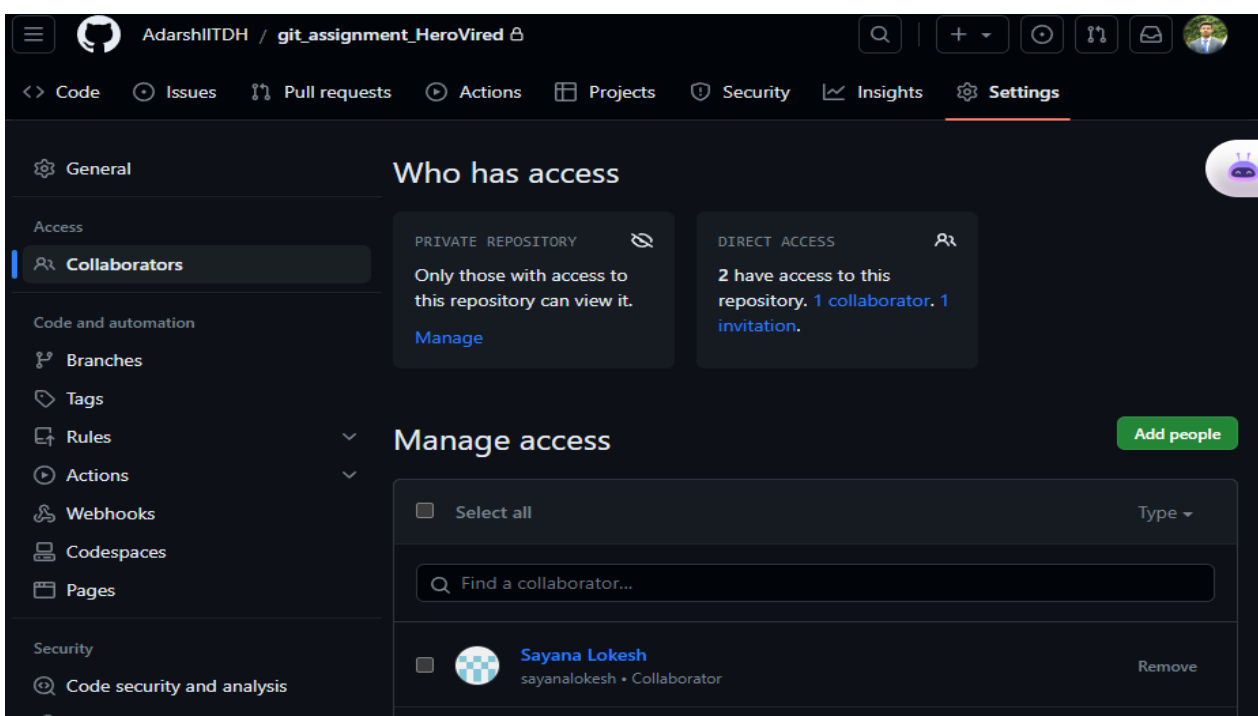
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git tag -a V1.0 -m "release of version-1 step c"

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AdarshIITDH/git_assignment_HeroVired.git
 8bd302b..72f6885  main -> main

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git push origin V1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 176 bytes | 176.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AdarshIITDH/git_assignment_HeroVired.git
 * [new tag]          V1.0 -> V1.0

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
```

- d) Add any of your classmates as collaborators.




The screenshot shows the GitHub repository settings for 'git_assignment_HeroVired' by user 'AdarshIITDH'. The 'Collaborators' tab is selected, displaying a list of collaborators. One collaborator, 'Sayana Lokesh' (sayanalokesh), is listed with a 'Remove' button. The page also shows 'Who has access' and 'Manage access' sections.

Who has access

- PRIVATE REPOSITORY**: Only those with access to this repository can view it. [Manage](#)
- DIRECT ACCESS**: 2 have access to this repository. 1 collaborator, 1 invitation.

Manage access

☐ Select all Type ▾

<input type="checkbox"/>	Avatar	Name	Role	Action
<input type="checkbox"/>		Sayana Lokesh	sayanalokesh • Collaborator	Remove

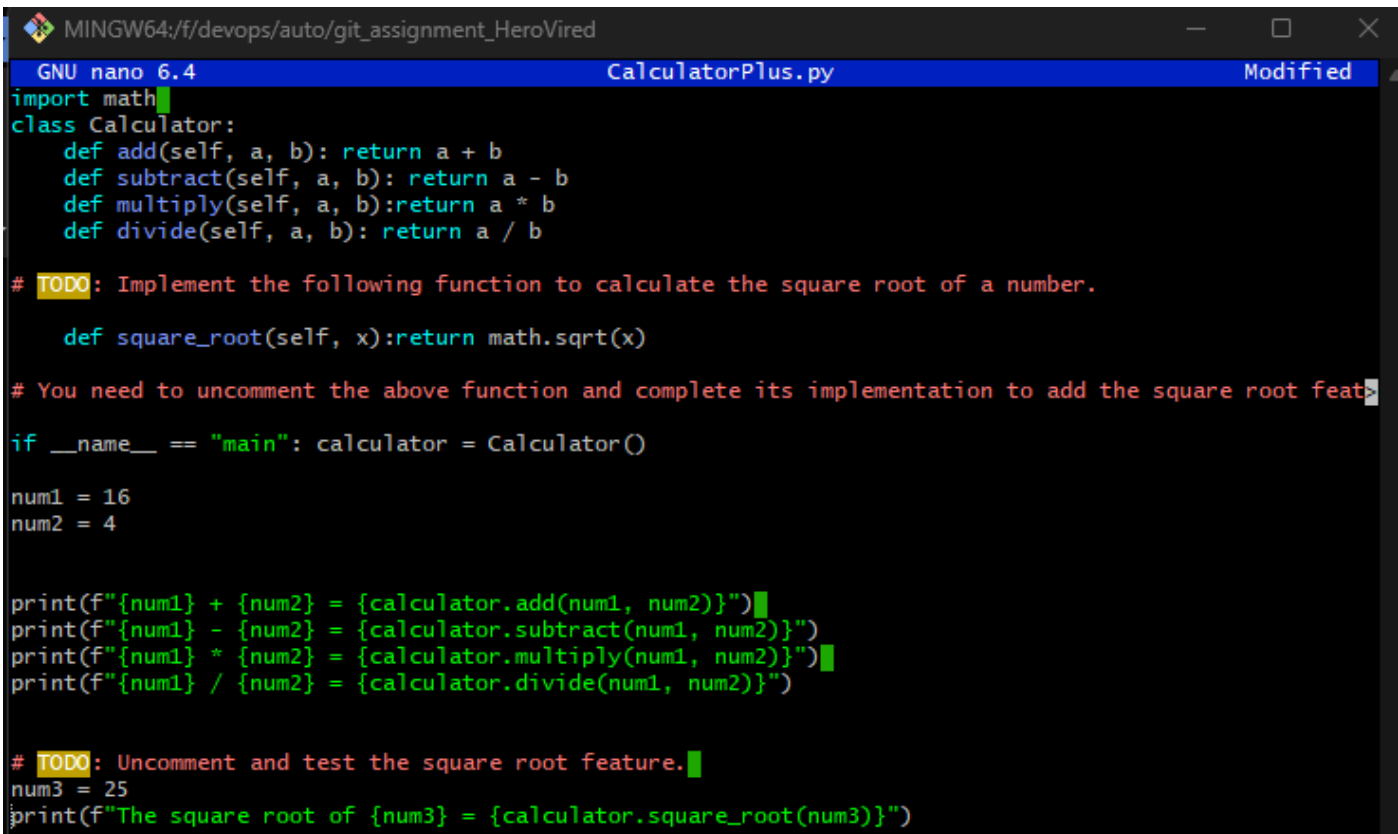
- e) Implement a feature by creating a new branch called 'feature/sqrt'.

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git checkout -b feature/sqrt
Switched to a new branch 'feature/sqrt'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ ls
CalculatorPlus.py  README.md

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ nano CalculatorPlus.py
```

- f) Add the 'sqrt' code into it.



```
MINGW64/f/devops/auto/git_assignment_HeroVired
GNU nano 6.4 CalculatorPlus.py Modified
import math
class Calculator:
    def add(self, a, b): return a + b
    def subtract(self, a, b): return a - b
    def multiply(self, a, b): return a * b
    def divide(self, a, b): return a / b

# TODO: Implement the following function to calculate the square root of a number.
    def square_root(self, x): return math.sqrt(x)

# You need to uncomment the above function and complete its implementation to add the square root feat

if __name__ == "main": calculator = Calculator()

num1 = 16
num2 = 4

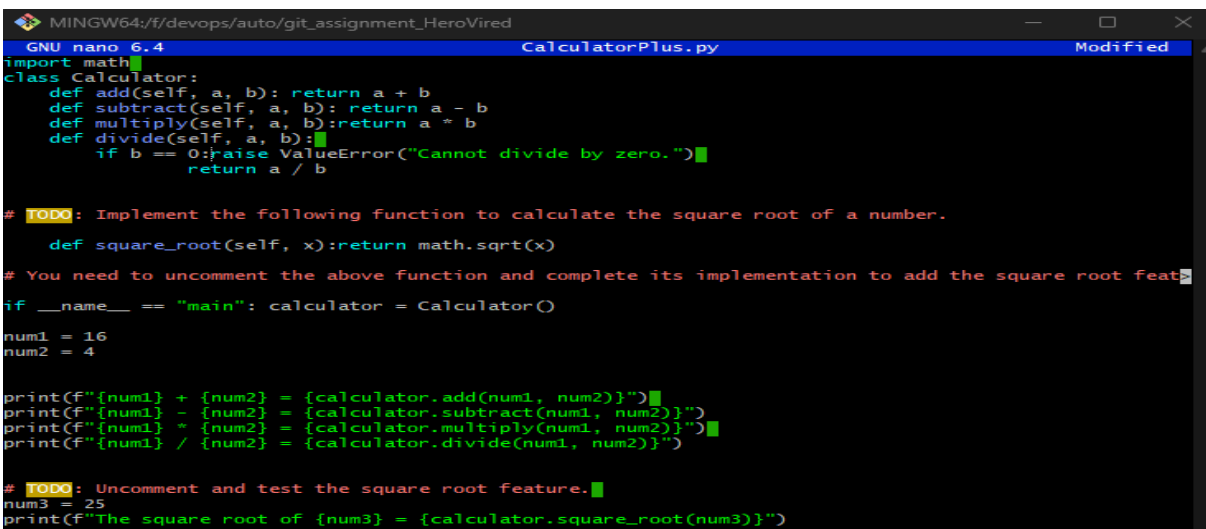
print(f"{num1} + {num2} = {calculator.add(num1, num2)}")
print(f"{num1} - {num2} = {calculator.subtract(num1, num2)}")
print(f"{num1} * {num2} = {calculator.multiply(num1, num2)}")
print(f"{num1} / {num2} = {calculator.divide(num1, num2)}")

# TODO: Uncomment and test the square root feature.
num3 = 25
print(f"The square root of {num3} = {calculator.square_root(num3)}")
```

- g) While you are working on this feature, imagine that one critical bug is reported in the main branch, and you need to switch back to the 'dev' branch, create fixes, and apply them while keeping your 'feature/sqrt' branch up-to-date. For this, you need to create

The bug fixation is in the divide function and the new function should be:

def divide(self, a, b): if b == 0: raise ValueError("Cannot divide by zero.") return a / b



```
MINGW64/f/devops/auto/git_assignment_HeroVired
GNU nano 6.4 CalculatorPlus.py Modified
import math
class Calculator:
    def add(self, a, b): return a + b
    def subtract(self, a, b): return a - b
    def multiply(self, a, b): return a * b
    def divide(self, a, b):
        if b == 0: raise ValueError("Cannot divide by zero.")
        return a / b

# TODO: Implement the following function to calculate the square root of a number.
    def square_root(self, x): return math.sqrt(x)

# You need to uncomment the above function and complete its implementation to add the square root feat

if __name__ == "main": calculator = Calculator()

num1 = 16
num2 = 4

print(f"{num1} + {num2} = {calculator.add(num1, num2)}")
print(f"{num1} - {num2} = {calculator.subtract(num1, num2)}")
print(f"{num1} * {num2} = {calculator.multiply(num1, num2)}")
print(f"{num1} / {num2} = {calculator.divide(num1, num2)}")

# TODO: Uncomment and test the square root feature.
num3 = 25
print(f"The square root of {num3} = {calculator.square_root(num3)}")
```

- h) After completing the feature implementation and ensuring that the application works correctly, create a pull request targeting the main branch.

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git checkout -b feature/sqrt
Switched to a new branch 'feature/sqrt'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ ls
CalculatorPlus.py  README.md

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ nano CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ git checkout dev
Switched to branch 'dev'
M      CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ nano CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git checkout feature/sqrt
Switched to branch 'feature/sqrt'
M      CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ nano CalculatorPlus.py

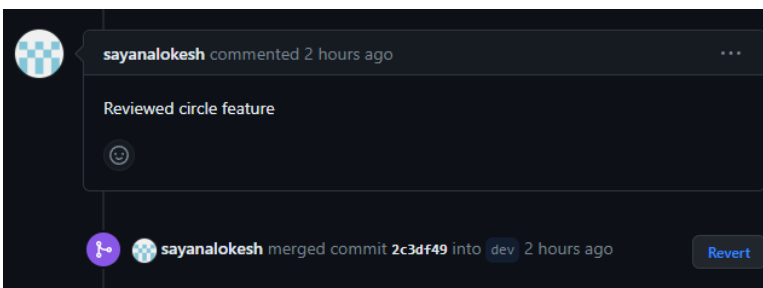
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ git add CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ git commit -m "step e bug is fixed in dev branch "
[feature/sqrt db2d2a0] step e bug is fixed in dev branch
1 file changed, 5 insertions(+), 5 deletions(-)

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$ git push origin feature/sqrt
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 451 bytes | 451.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/sqrt' on GitHub by visiting:
remote:   https://github.com/AdarshIITDH/git_assignment_HeroVired/pull/new/feature/sqrt
remote:
To github.com:AdarshIITDH/git_assignment_HeroVired.git
 * [new branch]      feature/sqrt -> feature/sqrt

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/sqrt)
$
```

- i. Request a code review from a team member and make any necessary improvements based on the review feedback.
- j. Once the code reviewer approves your pull request, merge the "feature/sqrt" branch into the 'dev' branch.



k. Finally, do the testing in the 'dev' branch itself and merge it into the 'main' branch and create a 'version 2' release.

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ nano CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ nano test_CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git add test_CalculatorPlus.py
warning: in the working copy of 'test_CalculatorPlus.py', LF will be replaced by CRLF the next time Git touches it

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git commit -m "sucessfully tested the calculator code"
[dev b7db741] sucessfully tested the calculator code
1 file changed, 40 insertions(+)
create mode 100644 test_CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git push origin dev
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 650 bytes | 650.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:AdarshIITDH/git_assignment_HeroVired.git
2fadcf6..b7db741 dev -> dev

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git merge dev
Updating 077b616..b7db741
Fast-forward
 geometry.py      | 19 ++++++
 test_CalculatorPlus.py | 40 ++++++
 2 files changed, 59 insertions(+)
 create mode 100644 geometry.py
 create mode 100644 test_CalculatorPlus.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AdarshIITDH/git_assignment_HeroVired.git
077b616..b7db741 main -> main
```

```
CalculatorPlus.py U  xyz.py U  test_CalculatorPlus.py U X
git_assignment_HeroVired > test_CalculatorPlus.py
1  import pytest
2  from CalculatorPlus import Calculator # Assuming your Calculator class is
3
4  def test_add():
5      calculator = Calculator()
6      assert calculator.add(2, 3) == 5
7      assert calculator.add(-1, 5) == 4
8      assert calculator.add(0, 0) == 0
9
10 def test_subtract():
11     calculator = Calculator()
12     assert calculator.subtract(5, 3) == 2
13     assert calculator.subtract(10, 7) == 3
14     assert calculator.subtract(0, 0) == 0
15
16 def test_multiply():
17     calculator = Calculator()
18     assert calculator.multiply(2, 3) == 6
19     assert calculator.multiply(5, -2) == -10
20     assert calculator.multiply(0, 0) == 0
21
22 def test_divide():
23     calculator = Calculator()
24     assert calculator.divide(6, 3) == 2
25     assert calculator.divide(10, 2) == 5
26     assert calculator.divide(0, 1) == 0

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS F:\devops\herovired\git_assignment_HeroVired> pytest
===== test session starts =====
platform win32 -- Python 3.10.7, pytest-7.4.0, pluggy-1.2.0
rootdir: F:\devops\herovired\git_assignment_HeroVired
plugins: Faker-15.3.2, csv-3.0.0
collected 5 items

test_CalculatorPlus.py ..... [100%]

===== 5 passed in 0.10s =====
PS F:\devops\herovired\git_assignment_HeroVired>
```

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git tag -a V2.0 -m "step-k version2 release"

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git push origin V2.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 174 bytes | 174.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AdarshIITDH/git_assignment_HeroVired.git
* [new tag]          V2.0 -> V2.0

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$
```


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. In the repository 'git_assignment_HeroVired', create a branch 'lfs'. Upload any large file whose size is over '200mb' and try to push this file into the repository.

```
adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (main)
$ git branch lfs

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (main)
$ git checkout lfs
Switched to branch 'lfs'

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ ls
README.md

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ ls
'Factory Demo Hands On Session _1-20230308_190229-Meeting Recording.mp4'
README.md

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ git lfs install
Updated Git hooks.
Git LFS initialized.

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ git lfs track "Factory Demo Hands On Session _1-20230308_190229-Meeting Recording.mp4"
Tracking "Factory Demo Hands On Session _1-20230308_190229-Meeting Recording.mp4"

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ git add Factory\ Demo\ Hands\ On\ Session\ _1-20230308_190229-Meeting\ Recording.mp4

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ git commit -m "large file of omniverse"
[lfs 3831b50] large file of omniverse
1 file changed, 3 insertions(+)
create mode 100644 Factory Demo Hands On Session _1-20230308_190229-Meeting Recording.mp4

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ git push origin lfs
Uploading LFS objects: 100% (1/1), 497 MB | 0 B/s, done.
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 459 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'lfs' on GitHub by visiting:
remote:   https://github.com/AdarshIITDH/git_assignment_HeroVired/pull/new/lfs
remote:
remote: To github.com:AdarshIITDH/git_assignment_HeroVired.git
* [new branch]      lfs -> lfs

adars@Adarsh MINGW64 /f/devops/herovired/git_assignment_HeroVired (lfs)
$ !
```

```
root@jarvis:/home/jarvis/Documents# git clone git@github.com:AdarshIITDH/git_assignment_HeroVired.git
Cloning into 'git_assignment_HeroVired'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 31 (delta 11), reused 13 (delta 2), pack-reused 0
Receiving objects: 100% (31/31), 7.12 KiB | 1.42 MiB/s, done.
Resolving deltas: 100% (11/11), done.
root@jarvis:/home/jarvis/Documents# cd git_assignment_HeroVired/
root@jarvis:/home/jarvis/Documents/git_assignment_HeroVired# ls
CalculatorPlus.py geometry.py README.md test_CalculatorPlus.py
root@jarvis:/home/jarvis/Documents/git_assignment_HeroVired# git checkout lfs
Branch 'lfs' set up to track remote branch 'lfs' from 'origin'.
Switched to a new branch 'lfs'
root@jarvis:/home/jarvis/Documents/git_assignment_HeroVired# git branch
* lfs
  main
root@jarvis:/home/jarvis/Documents/git_assignment_HeroVired# ls
'Factory Demo Hands On Session _1-20230308_190229-Meeting Recording.mp4' README.md
root@jarvis:/home/jarvis/Documents/git_assignment_HeroVired#
```

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.

a. Create a New Branch:

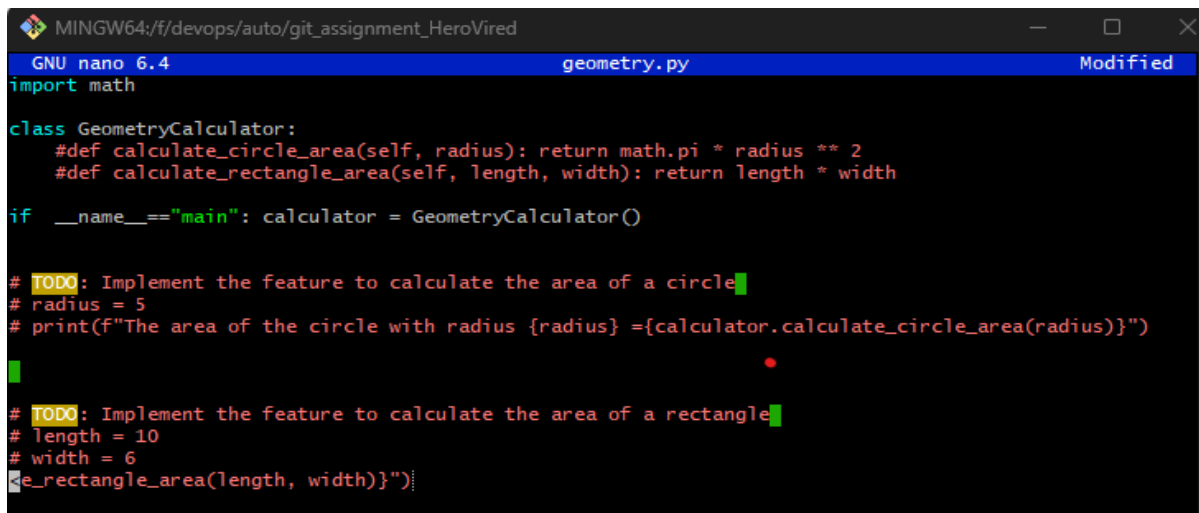
- Create a new branch named "feature/circle-area" to work on the circle-area feature

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (main)
$ git checkout -b geometry-calculator
Switched to a new branch 'geometry-calculator'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (geometry-calculator)
$ nano geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (geometry-calculator)
$ git checkout -b feature/circle-area
Switched to a new branch 'feature/circle-area'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ nano geometry.py
```



```
MINGW64:/f/devops/auto/git_assignment_HeroVired
GNU nano 6.4 geometry.py Modified
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()

# TODO: Implement the feature to calculate the area of a circle
# radius = 5
# print(f"The area of the circle with radius {radius} = {calculator.calculate_circle_area(radius)}")

# TODO: Implement the feature to calculate the area of a rectangle
# length = 10
# width = 6
# print(f"The area of the rectangle with length {length} and width {width} = {calculator.calculate_rectangle_area(length, width)}")
```

b. Stash Changes for Circle Area Feature:

- Before committing the changes, stash them using git stash to save the incomplete feature implementation.
- Verify that the working directory is clean

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ nano geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git stash save -u geometry.py
warning: in the working copy of 'geometry.py', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state On feature/circle-area: geometry.py
```



```
MINGW64:/f/devops/auto/git_assignment_HeroVired
GNU nano 6.4 geometry.py Modified
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()

# TODO: Implement the feature to calculate the area of a circle
radius = 5
print(f"The area of the circle with radius {radius} ={calculator.calculate_circle_area(radius)}")

# TODO: Implement the feature to calculate the area of a rectangle
# length = 10
# width = 6
# print(f"The area of the rectangle with length {length} and width {width} ={calculator.calculate_rect")
```

c. Create a New Branch for Rectangle Area Feature:

- Create a new branch named "feature/rectangle-area" to work on the rectangle area

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git checkout -b feature/rectangle-area
Switched to a new branch 'feature/rectangle-area'
```

d. Stash Changes for Rectangle Area Feature:

- Before committing the changes, stash them using git stash to save the incomplete feature implementation.
- Verify that the working directory is clean

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git checkout -b feature/rectangle-area
Switched to a new branch 'feature/rectangle-area'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ nano geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git stash save -u geometry.py
warning: in the working copy of 'geometry.py', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state On feature/rectangle-area: geometry.py
```

```
MINGW64:/f/devops/auto/dev
GNU nano 6.4 geometry.py Modified
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()

# TODO: Implement the feature to calculate the area of a circle
# radius = 5
# print(f"The area of the circle with radius {radius} ={calculator.calculate_ci

# TODO: Implement the feature to calculate the area of a rectangle
length = 10
width = 6
print(f"The area of the rectangle with length {length} and width {width} ={calc
```

e. Switch Back to Circle Area Branch:

- Switch back to the "feature/circle-area" branch to continue working on the circle area feature.
- Retrieve the stashed changes
- Complete the circle area feature implementation and save the changes.

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git checkout feature/circle-area
Switched to branch 'feature/circle-area'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git stash list
stash@{0}: On feature/rectangle-area: geometry.py
stash@{1}: On feature/circle-area: geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git stash pop 1
Already up to date.
On branch feature/circle-area
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    geometry.py

nothing added to commit but untracked files present (use "git add" to track)
Dropped refs/stash@{1} (45ebd1f064310bd940908129ba7f42c177205442)

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git add geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git commit -m "circle feature in feature/circle branch"
[feature/circle-area 24953c5] circle feature in feature/circle branch
 1 file changed, 19 insertions(+)
 create mode 100644 geometry.py
```

f. Commit and Push Circle Area Feature:

```
adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git push origin feature/circle-area
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 615 bytes | 615.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/circle-area' on GitHub by visiting:
remote:   https://github.com/AdarshIITDH/git_assignment_HeroVired/pull/new/feature/circle-area
remote:
To github.com:AdarshIITDH/git_assignment_HeroVired.git
 * [new branch]   feature/circle-area -> feature/circle-area
```

g. Switch Back to Rectangle Area Branch:

- Switch back to the "feature/rectangle-area" branch to continue working on the rectangle area feature.
- Retrieve the stashed changes
- Complete the rectangle area feature implementation and save the changes.

```

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/circle-area)
$ git checkout feature/rectangle-area
Switched to branch 'feature/rectangle-area'

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git stash pop 0
Already up to date.
On branch feature/rectangle-area
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    geometry.py

nothing added to commit but untracked files present (use "git add" to track)
Dropped refs/stash@{0} (43989b86049bc6b940928354e147ae0f3e1cfd30)

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git add geometry.py

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git commit -m "feature of rectangle area"
[feature/rectangle-area 9226f50] feature of rectangle area
1 file changed, 19 insertions(+)
create mode 100644 geometry.py

```

h. Commit and Push Rectangle Area Feature

```

adars@Adarsh MINGW64 /f/devops/auto/git_assignment_HeroVired (feature/rectangle-area)
$ git push origin feature/rectangle-area
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 610 bytes | 610.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/rectangle-area' on GitHub by visiting
:
remote:   https://github.com/AdarshIITDH/git_assignment_HeroVired/pull/new/feature/rectangle-area
remote:
To github.com:AdarshIITDH/git_assignment_HeroVired.git
* [new branch]   feature/rectangle-area -> feature/rectangle-area

```

- i. Create Pull Requests:
 - Create a pull request to the 'dev' branch.
- j. Review and Merge
 - Have another team member or reviewer review your pull requests.

Commits

dev

Commits on Aug 7, 2023

Merge pull request #7 from AdarshiITDH/feature/rectangle-area	Verified	2fadcf6	<>
sayanalokesh committed 1 hour ago			
Merge branch 'dev' into feature/rectangle-area	Verified	b79990b	<>
sayanalokesh committed 1 hour ago			
Merge pull request #6 from AdarshiITDH/feature/circle-area	Verified	2c3df49	<>
sayanalokesh committed 1 hour ago			
feature of rectangle area		9226f50	<>
AdarshiITDH committed 1 hour ago			
circle feature in feature/circle branch		24953c5	<>
AdarshiITDH committed 1 hour ago			
Merge pull request #2 from AdarshiITDH/dev	Verified	077b616	<>
SwapnashreeTripathy committed 3 hours ago			
Merge pull request #1 from AdarshiITDH/feature/sqrt	Verified	c2f6269	<>
sayanalokesh committed 3 hours ago			
step-e bug is fixed in dev branch now committing in feature branch		e3fbe46	<>
AdarshiITDH committed 5 hours ago			
step-b committing		72f6885	<>
AdarshiITDH committed 6 hours ago			
Initial commit	Verified	8bd302b	<>
AdarshiITDH committed 6 hours ago			

Newer Older

After receiving approval, merge both pull requests into the main branch.