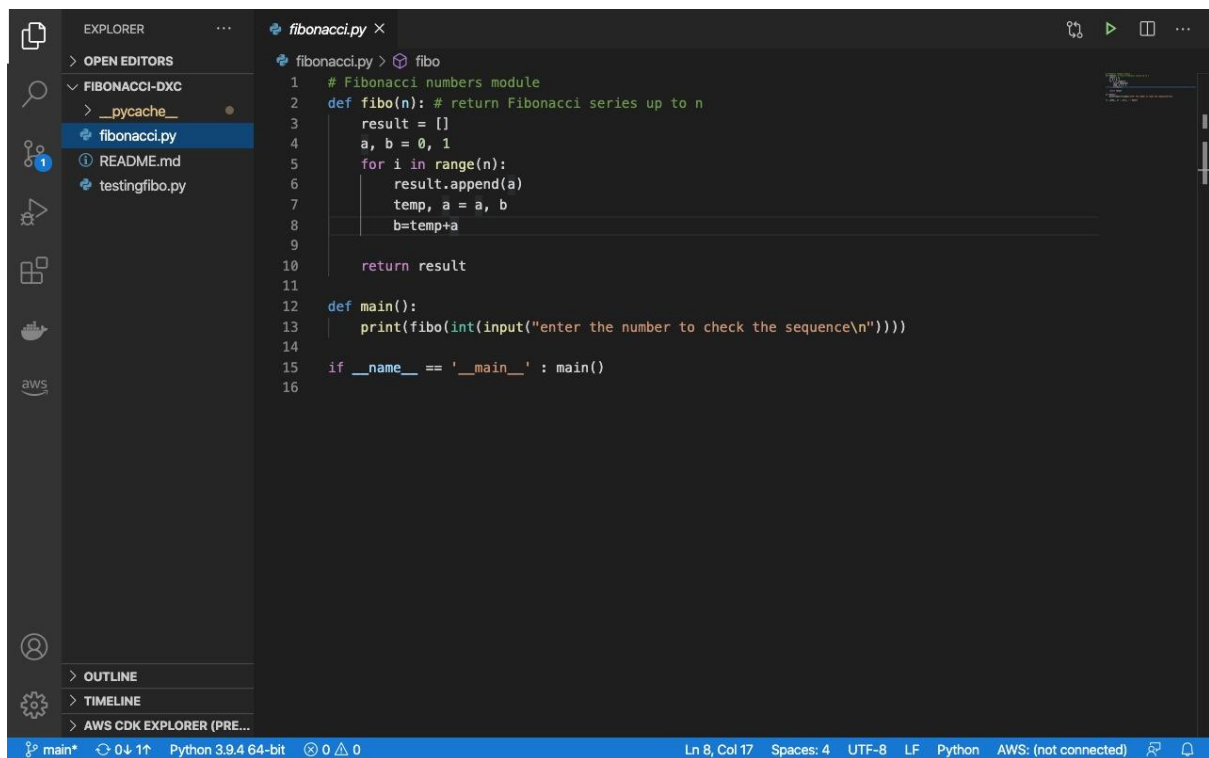


## ***PYTHON PROGRAM THAT DISPLAYS THE FIRST N FIBONACCI NUMBERS.***

The Fibonacci sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . . .

Each subsequent number is the sum of the previous two.

### **CODE FOR *FIBONACCI NUMBERS.***



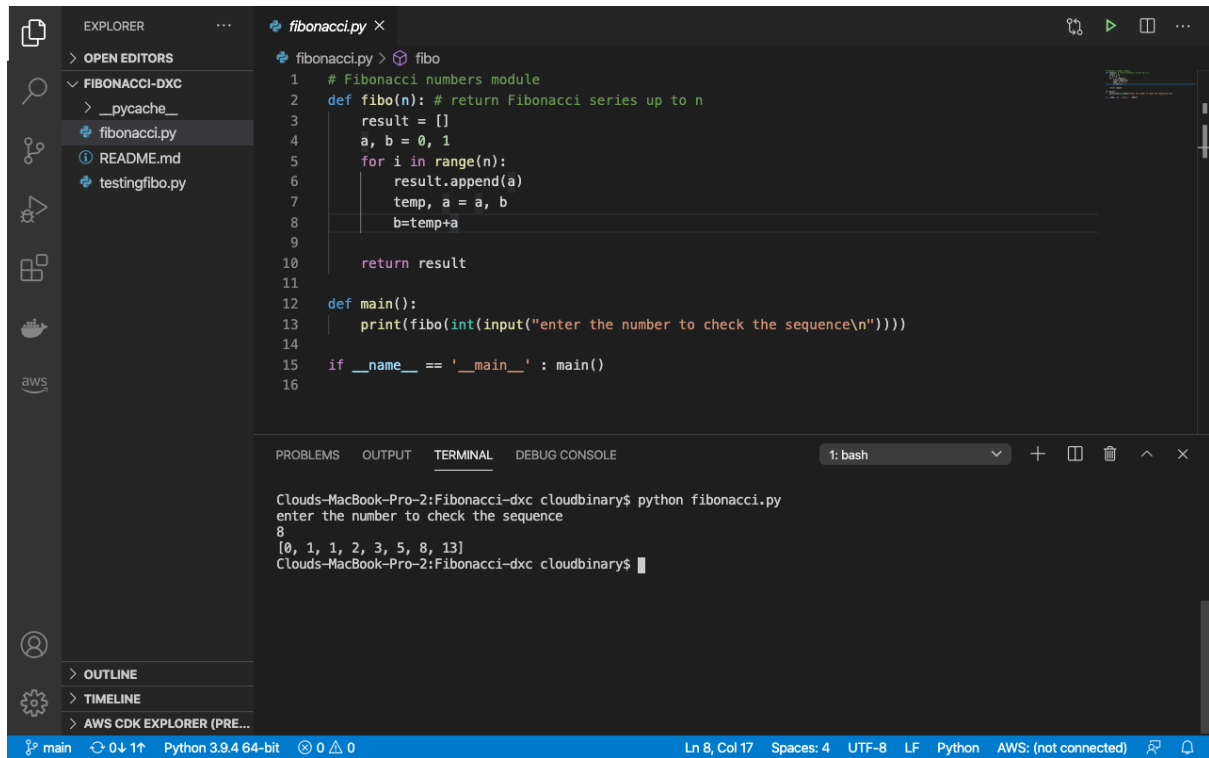
The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar is open, showing a file tree with 'FIBONACCI-DXC' containing 'fibonacci.py', 'README.md', and 'testingfibonacci.py'. The 'fibonacci.py' file is selected. The main editor area shows the following Python code:

```
1 # Fibonacci numbers module
2 def fibo(n): # return Fibonacci series up to n
3     result = []
4     a, b = 0, 1
5     for i in range(n):
6         result.append(a)
7         temp, a = a, b
8         b=temp+a
9
10    return result
11
12 def main():
13     print(fibo(int(input("enter the number to check the sequence\n"))))
14
15 if __name__ == '__main__': main()
16
```

The status bar at the bottom indicates the file is named 'main\*', the editor is at line 8, column 17, with 4 spaces, UTF-8 encoding, LF line endings, Python 3.9.4 64-bit, and AWS is not connected.

## EXECUTION OF *FIBONACCI* NUMBERS

Assuming n is set to 8 the following Fibonacci sequence should be displayed ->  
0,1,1,2,3,5,8,13



The screenshot shows a VS Code editor with a file explorer on the left containing 'FIBONACCI-DXC', '\_\_pycache\_\_', 'fibonacci.py', 'README.md', and 'testingfibonacci.py'. The main editor displays 'fibonacci.py' with the following code:

```
1 # Fibonacci numbers module
2 def fibo(n): # return Fibonacci series up to n
3     result = []
4     a, b = 0, 1
5     for i in range(n):
6         result.append(a)
7         temp, a = a, b
8         b=temp+a
9
10    return result
11
12 def main():
13     print(fibo(int(input("enter the number to check the sequence\n"))))
14
15 if __name__ == '__main__': main()
16
```

The bottom panel shows the 'TERMINAL' output:

```
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ python fibonacci.py
enter the number to check the sequence
8
[0, 1, 1, 2, 3, 5, 8, 13]
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$
```

The status bar at the bottom indicates 'Ln 8, Col 17', 'Spaces: 4', 'UTF-8', 'LF', 'Python', and 'AWS: (not connected)'.

EXECUTION OF THE PROGRAM,

Python <filename>.py

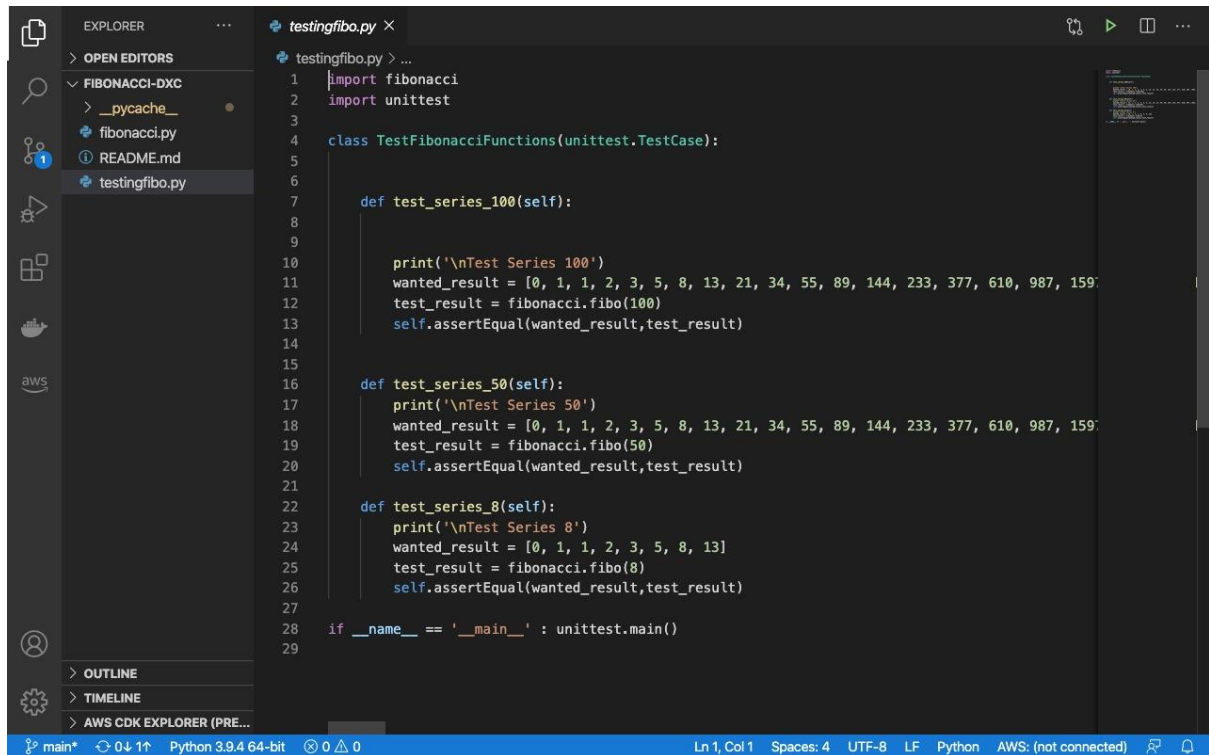
Python fibonacci.py

Enter the number to check the sequence

8

[0, 1, 1, 2, 3, 5, 8, 13]

# UNIT TESTING



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'FIBONACCI-DXC' with files: '\_\_pycache\_\_', 'fibonacci.py', 'README.md', and 'testingfibo.py'. The code editor shows the content of 'testingfibo.py'.

```
1 import fibonacci
2 import unittest
3
4 class TestFibonacciFunctions(unittest.TestCase):
5
6
7     def test_series_100(self):
8
9
10         print('\nTest Series 100')
11         wanted_result = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597]
12         test_result = fibonacci.fibo(100)
13         self.assertEqual(wanted_result, test_result)
14
15
16     def test_series_50(self):
17         print('\nTest Series 50')
18         wanted_result = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597]
19         test_result = fibonacci.fibo(50)
20         self.assertEqual(wanted_result, test_result)
21
22
23     def test_series_8(self):
24         print('\nTest Series 8')
25         wanted_result = [0, 1, 1, 2, 3, 5, 8, 13]
26         test_result = fibonacci.fibo(8)
27         self.assertEqual(wanted_result, test_result)
28
29 if __name__ == '__main__': unittest.main()
```

UNIT TESTING FOR FABONICCI NUMBERS IT IS TESTED FOR THE SERIES OF 100, 50 AND 8 THE TEST SERIES RAN SUCCESSFULLY. FOR THE THREE TEST CASES.

FOR THE EXECUTION OF UNIT TESTING SYNTAX IS : `python <testfile>.py`

**python testingfibo.py**

**Test Series 100**

.

**Test Series 50**

.

**Test Series 8**

.

**Ran 3 tests in 0.003s**

**OK**

```
1 import fibonacci
2 import unittest
3
4 class TestFibonacciFunctions(unittest.TestCase):
5
6     def test_series_100(self):
7
8         print('\nTest Series 100')
9         wanted_result = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597]
10        test_result = fibonacci.fibo(100)
11        self.assertEqual(wanted_result, test_result)
12
13    def test_series_50(self):
14        print('\nTest Series 50')
15        wanted_result = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597]
16
17    def test_series_8(self):
18        print('\nTest Series 8')
```

```
Clouds-MacBook-Pro-2: Fibonacci-dxc ccloudbinary$ python testingfibo.py
Test Series 100
Test Series 50
Test Series 8
Ran 3 tests in 0.001s
OK
Clouds-MacBook-Pro-2: Fibonacci-dxc ccloudbinary$
```

## CREATING NEW GITHUB REPOSITORY

elsewhere? [Import a repository.](#)

Owner \* Wilson-Devops / Repository name \* Fibonacci@dxc

Great repository names are short, lowercase, and contain only alphanumeric characters and hyphens. Your new repository will be created as Fibonacci-dxc. Iterate-journey?

Description (optional)  
Python program that displays the first n Fibonacci numbers.

☒ **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:  
Skip this step if you're importing an existing repository.

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

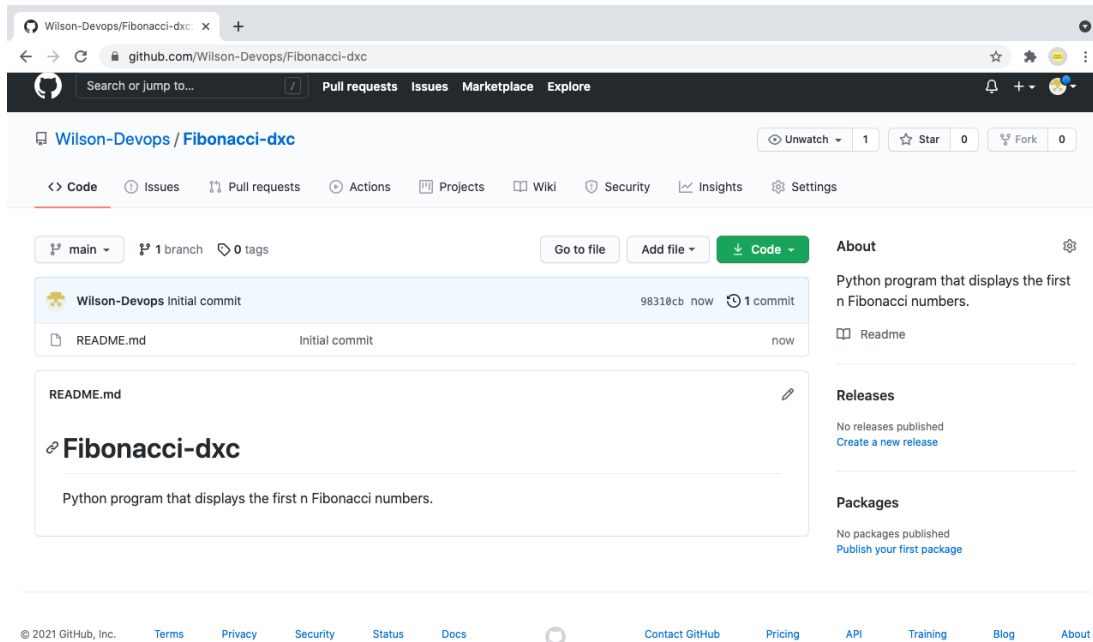
☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

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

[Create repository](#)

# NEW REPO

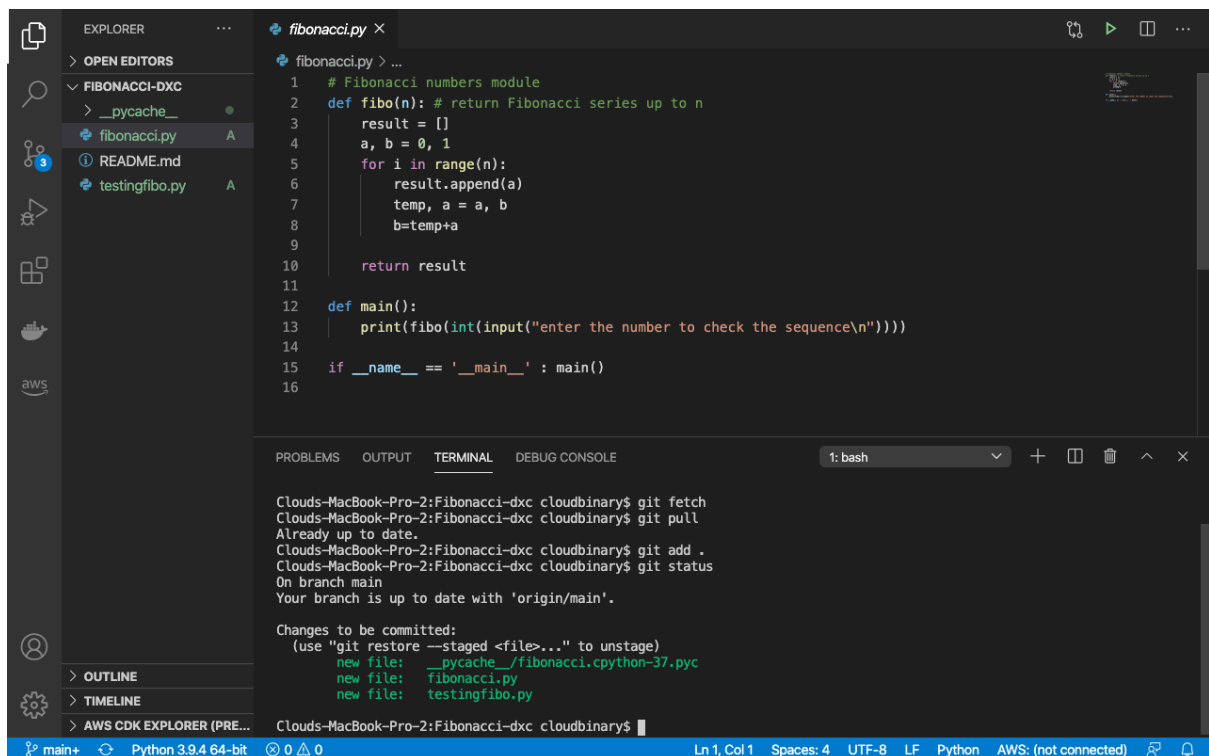


**Git URL:** <https://github.com/Wilson-Devops/Fibonacci-dxc.git>

# CLONING INTO LOCAL REPOSITORY

```
[Clouds-MacBook-Pro-2:wilson@dxc cloudbinary$ git clone https://github.com/Wilson-Devops/Fibonacci-dxc.git
Cloning into 'Fibonacci-dxc'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Clouds-MacBook-Pro-2:wilson@dxc cloudbinary$
```

## ADDING CODE



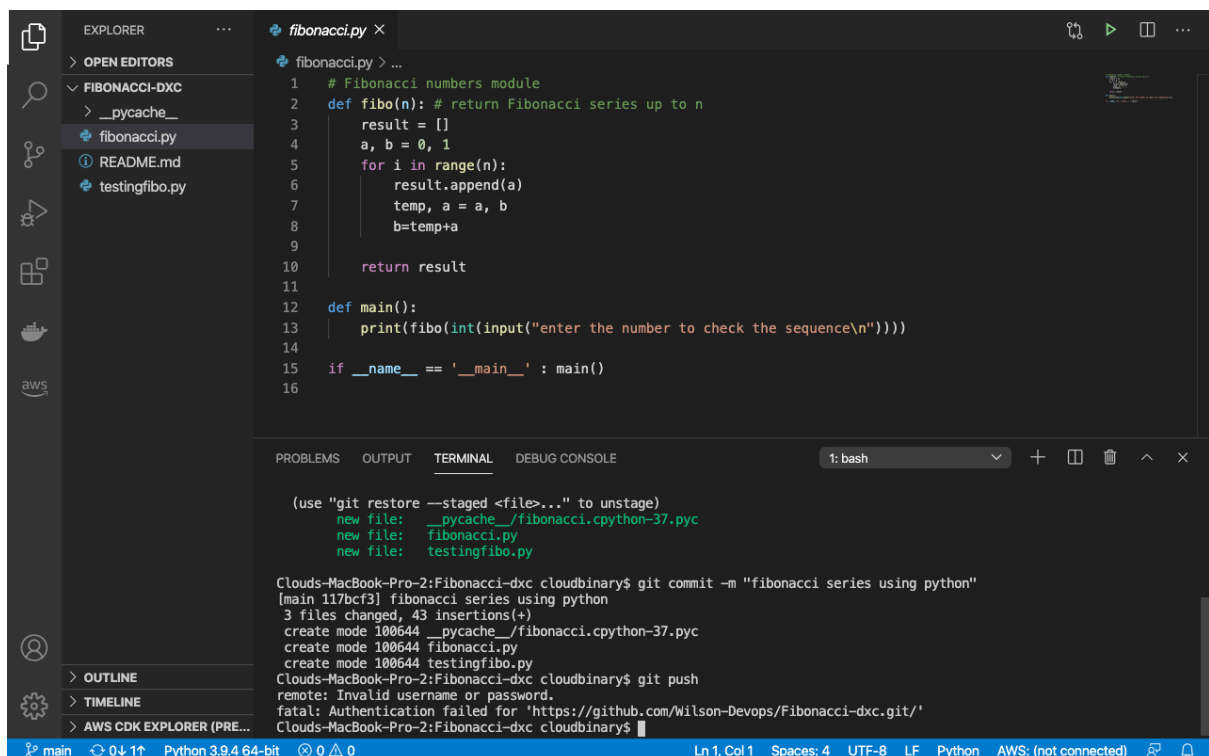
The screenshot shows the VS Code interface with the Explorer sidebar on the left. The 'FIBONACCI-DXC' folder is expanded, showing files: `__pycache__`, `fibonacci.py`, `README.md`, and `testingfibonacci.py`. The `fibonacci.py` file is open in the editor, showing a Python script for calculating Fibonacci numbers. The script includes a `def fibo(n):` function and a `def main():` function that takes user input. The terminal at the bottom shows the following commands and output:

```
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git fetch
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git pull
Already up to date.
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git add .
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   __pycache__/fibonacci.cpython-37.pyc
    new file:   fibonacci.py
    new file:   testingfibonacci.py

Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$
```

## COMMITTING CHANGES AND PUSHING TO REMOTE REPO



The screenshot shows the VS Code interface with the same files as the previous image. The terminal at the bottom shows the following commands and output:

```
(use "git restore --staged <file>..." to unstage)
  new file:   __pycache__/fibonacci.cpython-37.pyc
  new file:   fibonacci.py
  new file:   testingfibonacci.py

Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git commit -m "fibonacci series using python"
[main 117bcf3] fibonacci series using python
3 files changed, 43 insertions(+)
create mode 100644 __pycache__/fibonacci.cpython-37.pyc
create mode 100644 fibonacci.py
create mode 100644 testingfibonacci.py
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$ git push
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/Wilson-Devops/Fibonacci-dxc.git/'
Clouds-MacBook-Pro-2: Fibonacci-dxc cloudbinary$
```

Git URL: <https://github.com/Wilson-Devops/Fibonacci-dxc.git>