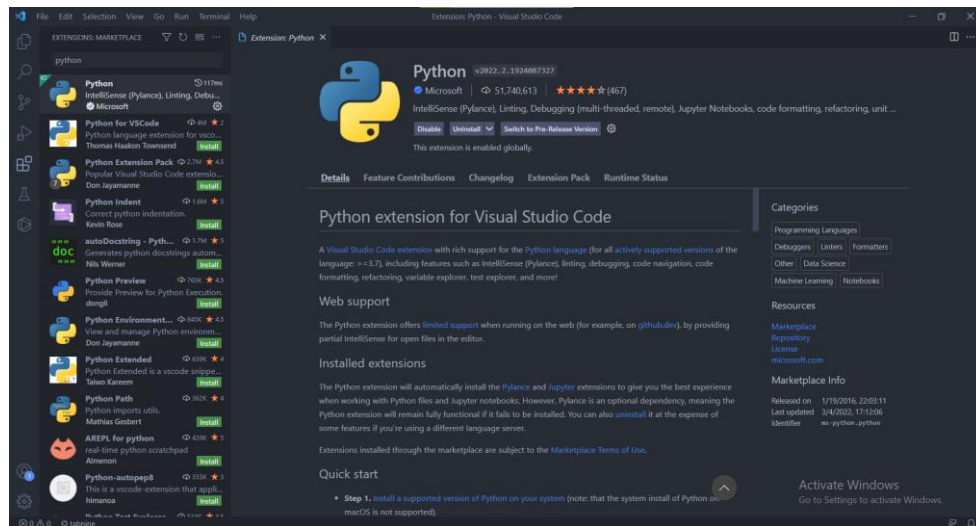


6206021611125

ชรณศร เมตตา

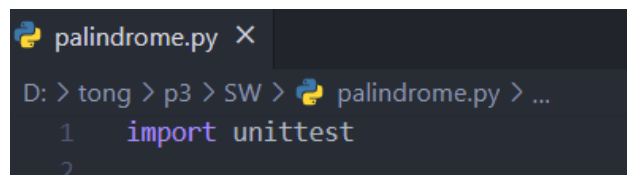
1.python

เริ่มจากการติดตั้ง Extension Python

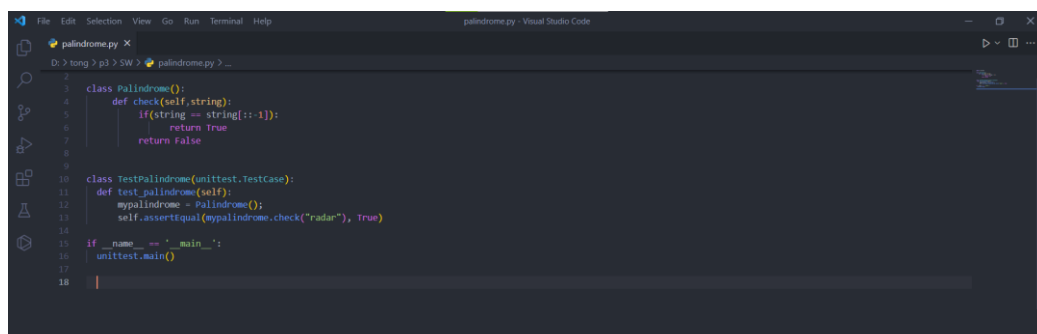


จากนั้นทำการสร้างไฟล์ขึ้นมา โดยจะใช้ชื่อว่า “palindrome.py”

แล้วทำการ import module ที่จะใช้ด้วยคำสั่ง “import unittest”



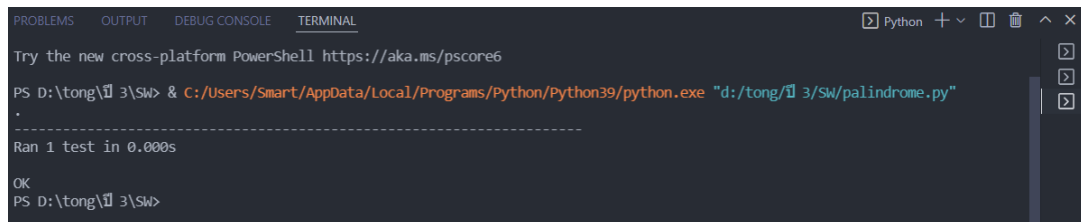
เริ่มทำการเขียนโค้ด โดยที่ฟังก์ชันทดสอบจะต้องอยู่ภายใน class และชื่อฟังก์ชันต้องขึ้นต้นด้วย “test”



ใช้ Method “assertEqual” เพื่อส่งค่าไปยังฟังก์ชัน check

```
self.assertEqual(mypalindrome.check("radar"), True)
```

รันแล้วจะได้ผลลัพธ์แบบนี้



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\tong\ปี 3\Sw> & C:/Users/Smart/AppData/Local/Programs/Python/Python39/python.exe "d:/tong/ปี 3/Sw/palindrome.py"
.
-----
Ran 1 test in 0.000s
OK
PS D:\tong\ปี 3\Sw>
```

โค้ด

```
import unittest

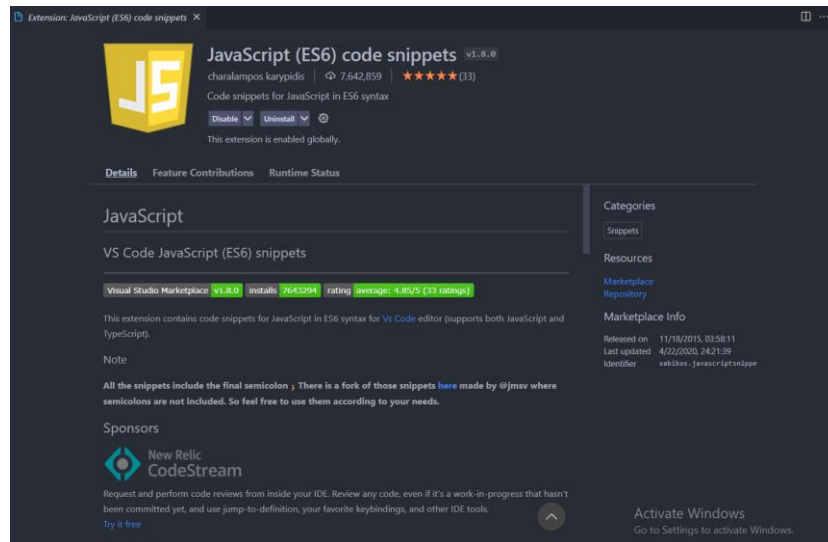
class Palindrome():
    def check(self,string):
        if(string == string[::-1]):
            return True
        return False

class TestPalindrome(unittest.TestCase):
    def test_palindrome(self):
        mypalindrome = Palindrome();
        self.assertEqual(mypalindrome.check("radar"), True)

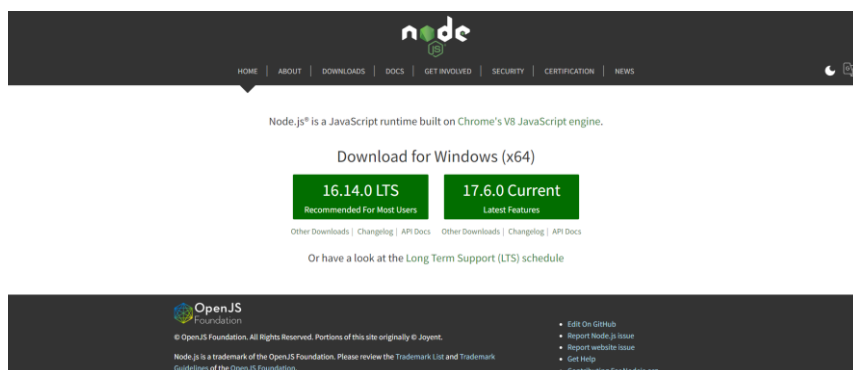
if __name__ == '__main__':
    unittest.main()
```

2. JavaScript

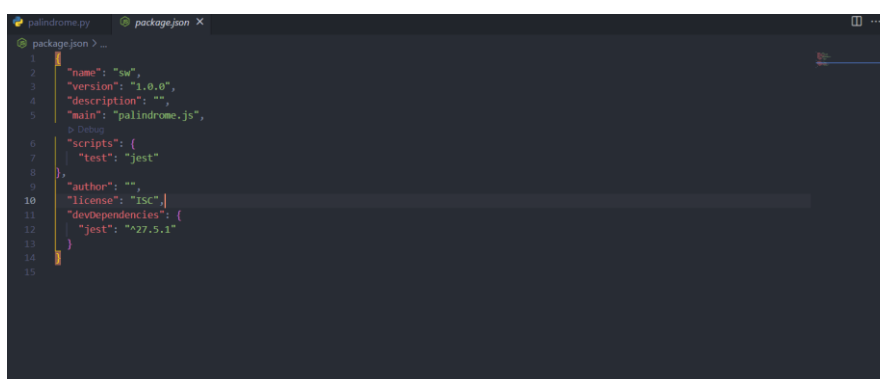
เริ่มจากการติดตั้ง Extension JavaScript



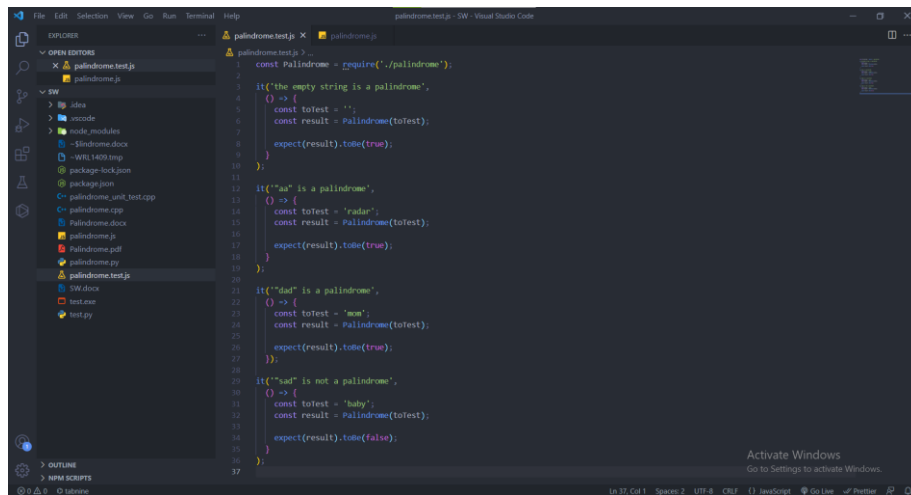
จากนั้นทำการติดตั้ง NodeJS



ทำการ Install platform Jest ด้วยคำสั่ง “npm install --save-dev jest” จะได้ไฟล์ชื่อ “package.json”

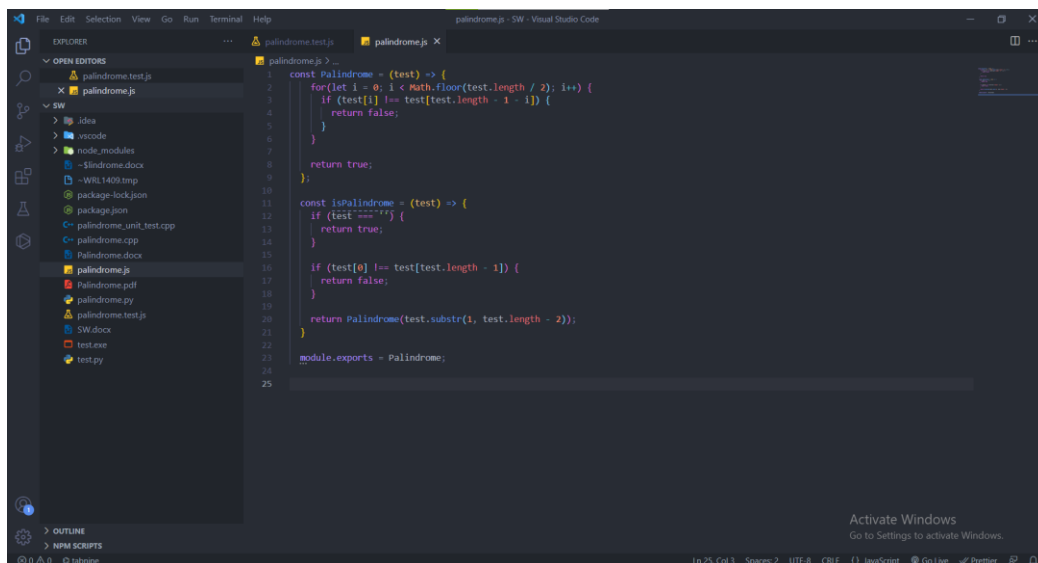


ทำการสร้างไฟล์ Test ขึ้นมา โดยเราจะใช้ชื่อว่า palindrome.test.js



```
1 const Palindrome = require('./palindrome');
2
3 it('the empty string is a palindrome',
4   () => {
5     const toTest = '';
6     const result = Palindrome(toTest);
7     expect(result).toBe(true);
8   });
9
10 it('aa is a palindrome',
11   () => {
12     const toTest = 'aa';
13     const result = Palindrome(toTest);
14     expect(result).toBe(true);
15   });
16
17 it('radar is a palindrome',
18   () => {
19     const toTest = 'radar';
20     const result = Palindrome(toTest);
21     expect(result).toBe(true);
22   });
23
24 it('dad is a palindrome',
25   () => {
26     const toTest = 'dad';
27     const result = Palindrome(toTest);
28     expect(result).toBe(true);
29   });
30
31 it('sad is not a palindrome',
32   () => {
33     const toTest = 'sad';
34     const result = Palindrome(toTest);
35     expect(result).toBe(false);
36   });
37
```

ทำการสร้างไฟล์ palindrome.js ขึ้นมาเพื่อเขียนฟังก์ชัน



```
1 const Palindrome = (test) => {
2   for(let i = 0; i < Math.Floor(test.length / 2); i++) {
3     if(test[i] !== test[test.length - 1 - i]) {
4       return false;
5     }
6   }
7   return true;
8 };
9
10 const isPalindrome = (test) => {
11   if (test === '') {
12     return true;
13   }
14   if (test[0] !== test[test.length - 1]) {
15     return false;
16   }
17   return Palindrome(test.substr(1, test.length - 2));
18 }
19
20 module.exports = Palindrome;
21
22
23
24
25
```

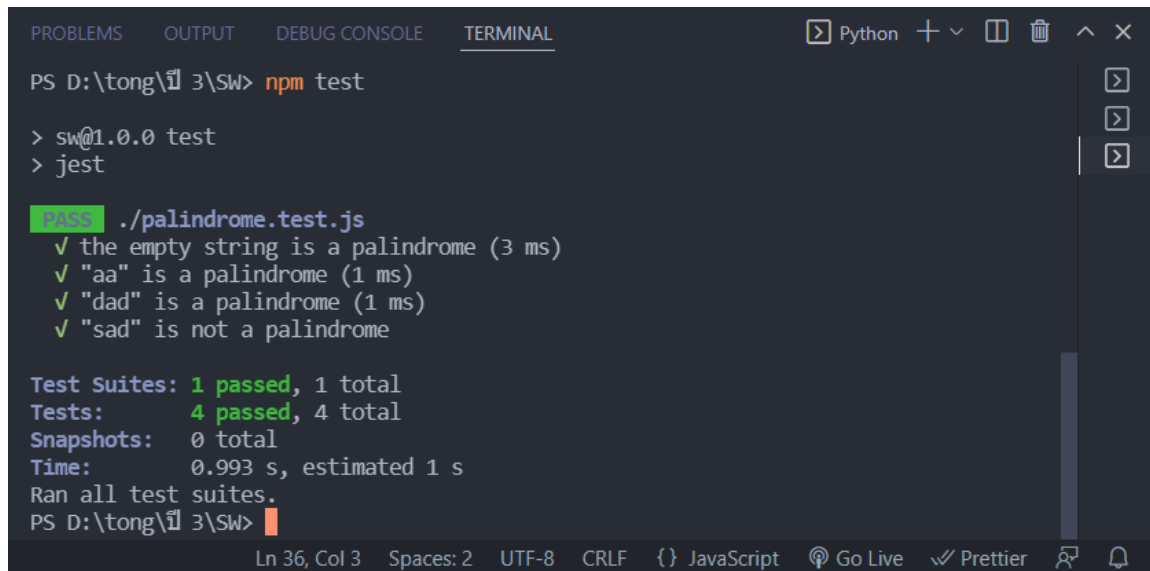
เราจะทำการส่งฟังก์ชันออกมาจาก palindrome.js ผ่านทางออบเจ็ค “module.exports”

```
module.exports = Palindrome;
```

จากนั้นเราใช้ฟังก์ชัน require นำเข้าฟังก์ชันจาก palindrome.js เพื่อนำมาใช้งานใน palindrome.test.js

```
const Palindrome = require('./palindrome');
```

ทำการรันด้วยคำสั่ง “npm test”



The screenshot shows a VS Code terminal window with the following content:

```
PS D:\tong\ปี 3\SW> npm test

> sw@1.0.0 test
> jest

PASS ./palindrome.test.js
  ✓ the empty string is a palindrome (3 ms)
  ✓ "aa" is a palindrome (1 ms)
  ✓ "dad" is a palindrome (1 ms)
  ✓ "sad" is not a palindrome

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.993 s, estimated 1 s
Ran all test suites.
PS D:\tong\ปี 3\SW>
```

The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The status bar at the bottom shows 'Ln 36, Col 3', 'Spaces: 2', 'UTF-8', 'CRLF', '{ } JavaScript', 'Go Live', 'Prettier', and icons for search and notifications.

โค้ด

- palindrome.test.js

```
const Palindrome = require('./palindrome');

it('the empty string is a palindrome',
  () => {
    const toTest = '';
    const result = Palindrome(toTest);

    expect(result).toBe(true);
  }
);

it('"aa" is a palindrome',
  () => {
    const toTest = 'radar';
    const result = Palindrome(toTest);

    expect(result).toBe(true);
  }
);

it('"dad" is a palindrome',
  () => {
    const toTest = 'mom';
    const result = Palindrome(toTest);
```

```

    expect(result).toBe(true);
  });

it('"sad" is not a palindrome',
  () => {
    const toTest = 'baby';
    const result = Palindrome(toTest);

    expect(result).toBe(false);
  }
);

```

- palindrome.js

```

const Palindrome = (test) => {
  for(let i = 0; i < Math.floor(test.length / 2); i++) {
    if (test[i] !== test[test.length - 1 - i]) {
      return false;
    }
  }

  return true;
};

const isPalindrome = (test) => {
  if (test === '') {
    return true;
  }

  if (test[0] !== test[test.length - 1]) {
    return false;
  }

  return Palindrome(test.substr(1, test.length - 2));
}

module.exports = Palindrome;

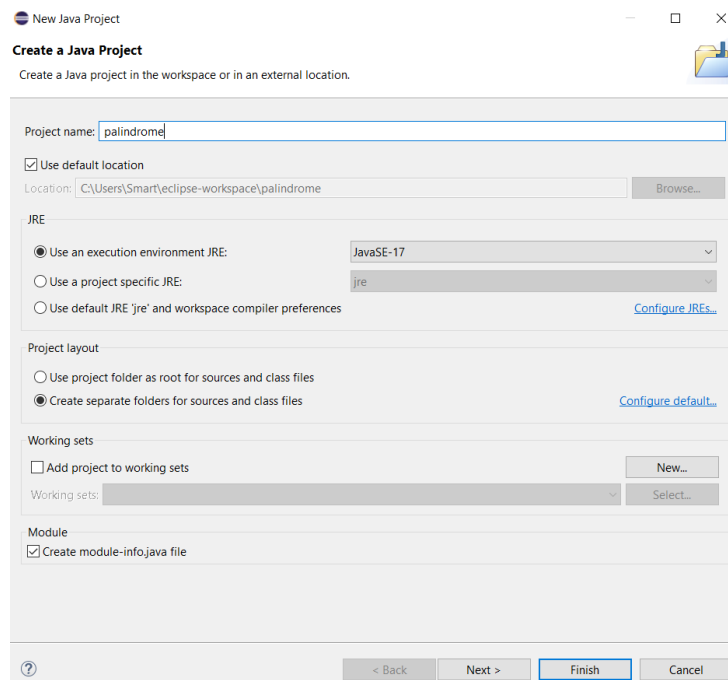
```

3. Java

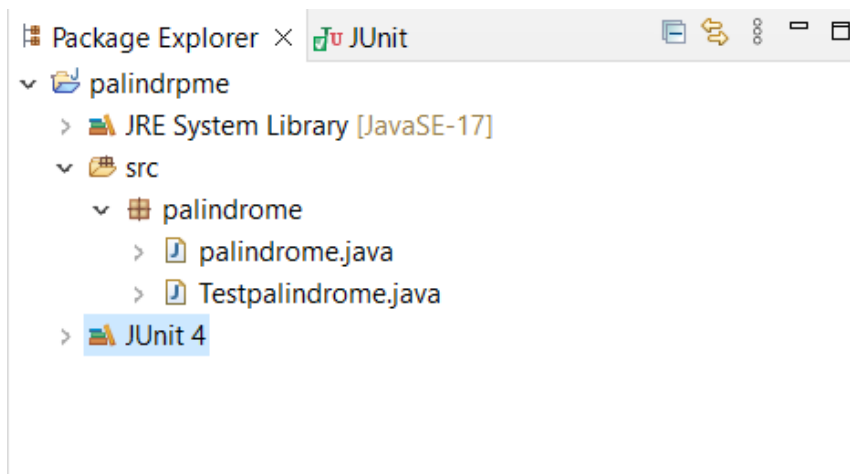
โดยทดสอบผ่านโปรแกรม Eclipse IDE

เริ่มจากการติดตั้ง Library Junit บน Eclipse IDE ก่อน

จากนั้นทำการสร้าง Java project ขึ้นมา



ทำการสร้างไฟล์ขึ้นมา 2 ไฟล์ คือ Testpalindrome.java และ palindrome.java

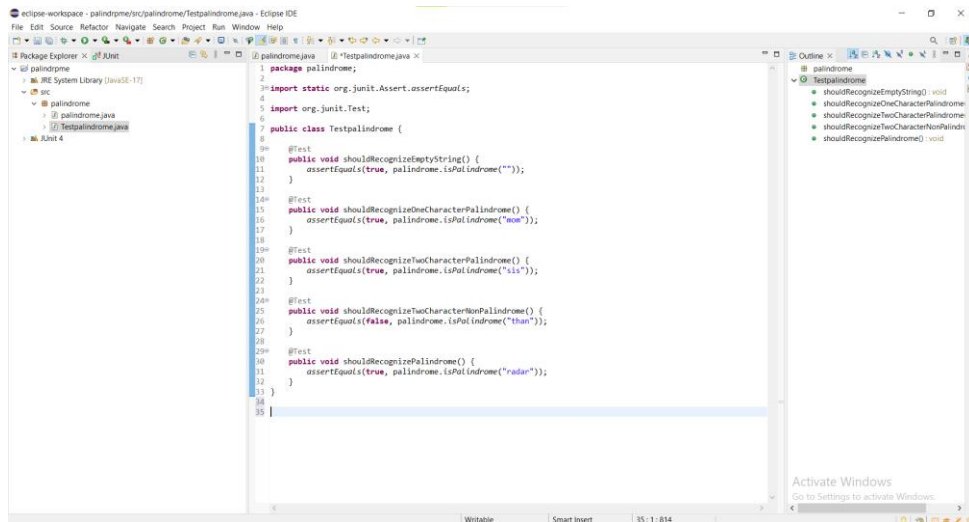


จากนั้นทำการ import “org.junit.Assert.assertEquals” และ “org.junit.Test” ในไฟล์ Test

```
import static org.junit.Assert.assertEquals;

import org.junit.Test;
```

แล้วทำการเขียนโค้ดในไฟล์ Testpalindrom.java

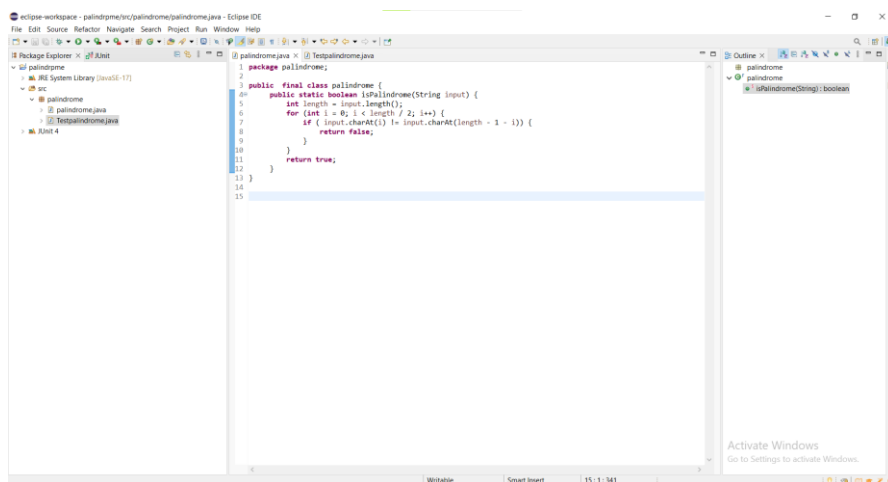


เราจะใช้ @Test เพื่อบอกให้ตัว Junit ทราบว่า method นี้จะเป็น unit test

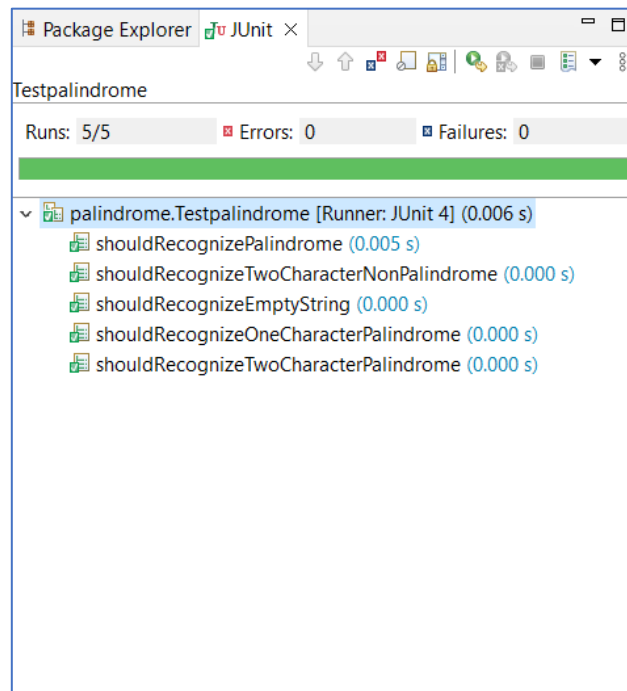
ใช้ Method “assertEquals” เพื่อส่งค่าไปยังฟังก์ชัน isPalindrome ภายในไฟล์ palindrom.java

```
assertEquals(true, palindrome.isPalindrome(""));
```

จากนั้นทำการเขียนโค้ดในไฟล์ palindrome.java



เมื่อรันออกมาจะได้หน้าต่างแบบนี้



โค้ด

- Testpalindrome.java

```
package palindrome;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class Testpalindrome {

    @Test
    public void shouldRecognizeEmptyString(){
        assertEquals(true, palindrome.isPalindrome(""));
    }

    @Test
    public void shouldRecognizeOneCharacterPalindrome(){
        assertEquals(true, palindrome.isPalindrome("mom"));
    }

    @Test
    public void shouldRecognizeTwoCharacterPalindrome(){
        assertEquals(true, palindrome.isPalindrome("sis"));
    }

    @Test
```

```

public void shouldRecognizeTwoCharacterNonPalindrome(){
    assertEquals(false, palindrome.isPalindrome("than"));
}

@Test
public void shouldRecognizePalindrome(){
    assertEquals(true, palindrome.isPalindrome("radar"));
}
}

```

- palindrome.java

```

package palindrome;

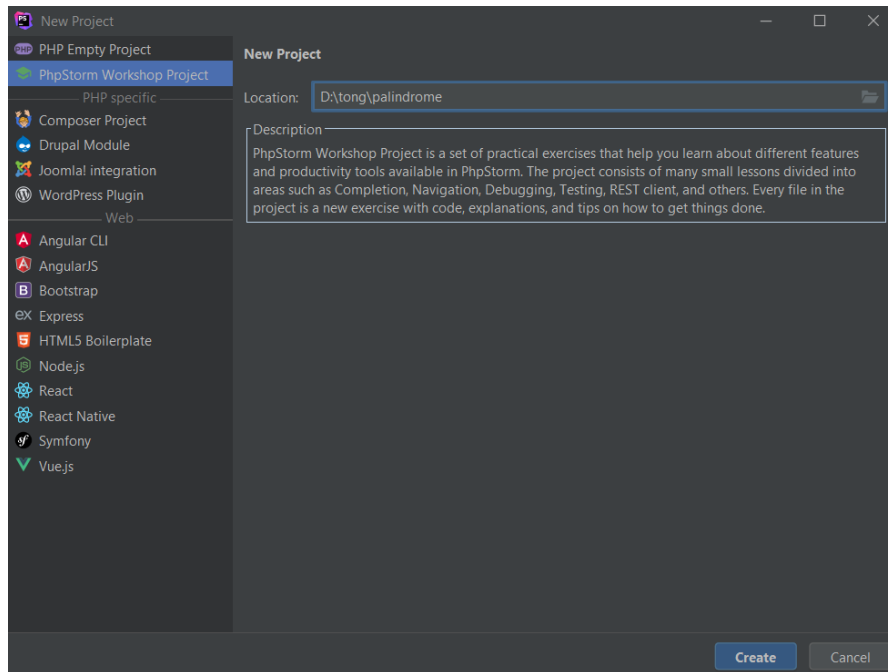
public final class palindrome {
    public static boolean isPalindrome(String input){
        int length=input.length();
        for(int i=0; i < length/2; i++){
            if(input.charAt(i)!=input.charAt(length-1-i)){
                return false;
            }
        }
        return true;
    }
}

```

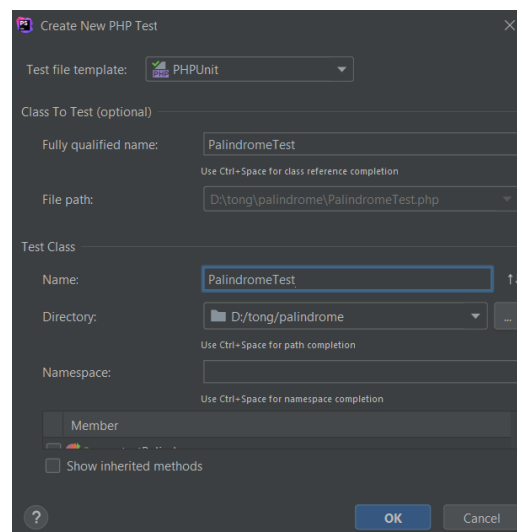
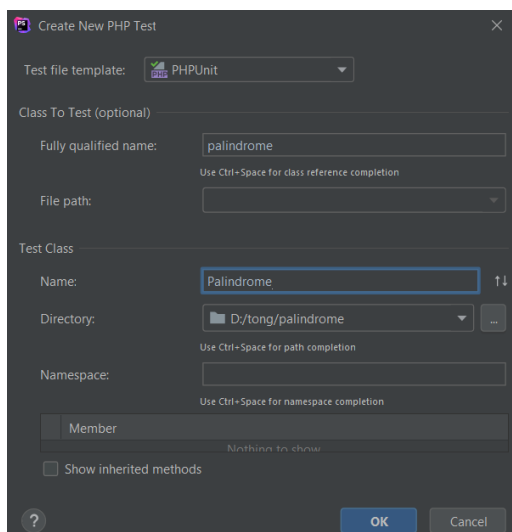
4.php

โดยทดสอบผ่านโปรแกรม PhpStorm

เริ่มแรกทำการสร้าง project ขึ้นมา



จากนั้นทำการสร้างไฟล์ Palindrome และ PalindromeTest

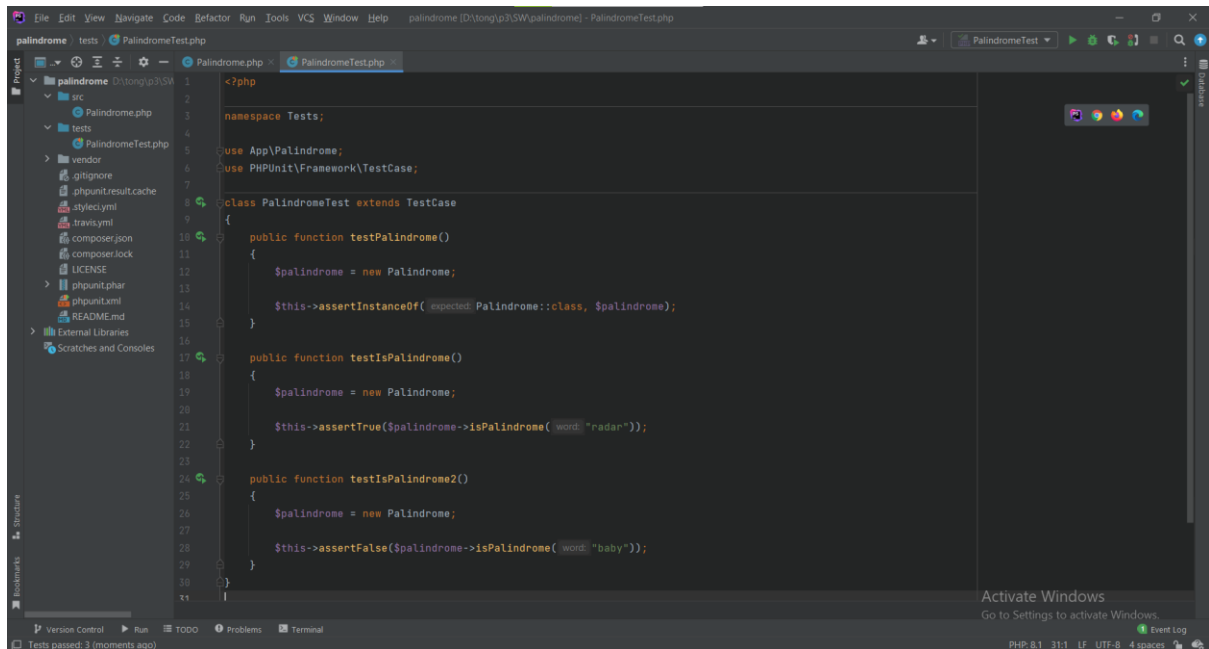


ทำการติดตั้ง Library Phpunit บน PhpStorm

ในไฟล์ PalindromeTest.php ทำการเรียกใช้

```
5 use App\Palindrome;  
6 use PHPUnit\Framework\TestCase;
```

จากนั้นทำการเขียนโค้ดภายในคลาส



ใช้ Method “assertInstanceOf”

```
$this->assertInstanceOf( expected: Palindrome::class, $palindrome);
```

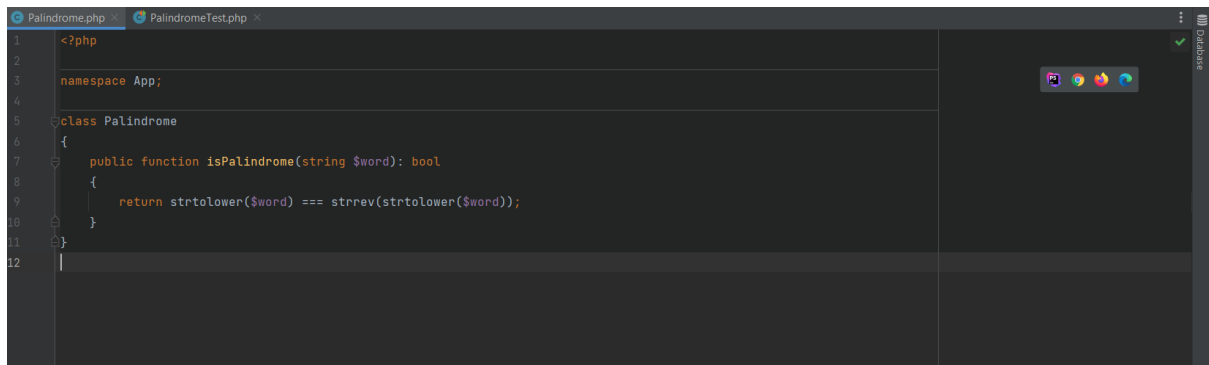
ใช้ Method “assertTrue” เพื่อตรวจสอบว่าค่าที่ส่งไปเป็นจริง

```
$this->assertTrue($palindrome->isPalindrome( word: "radar"));
```

ใช้ Method “assertFalse” เพื่อตรวจสอบว่าค่าที่ส่งไปเป็นเท็จ

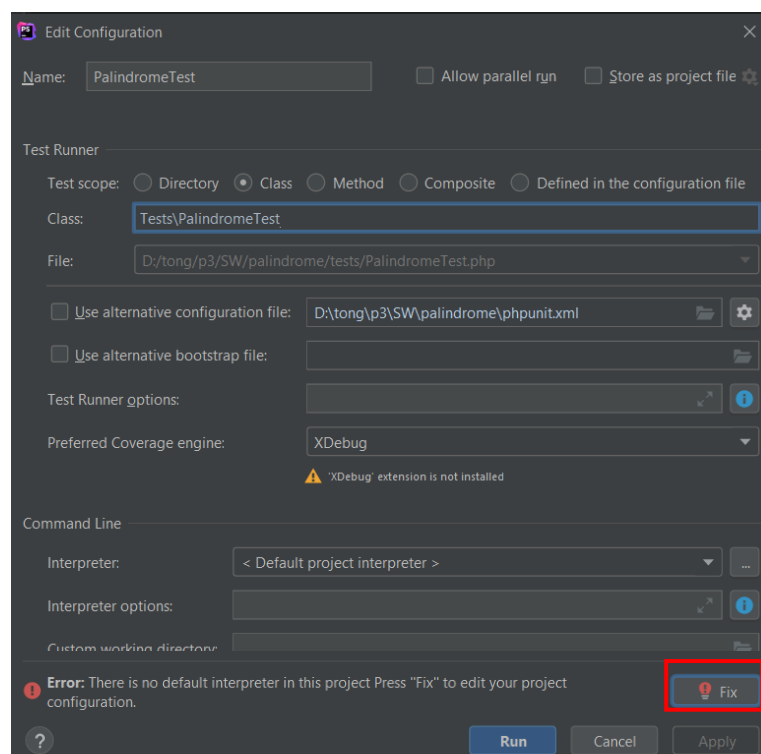
```
$this->assertFalse($palindrome->isPalindrome( word: "baby"));
```

จากนั้นทำการเขียนโค้ดภายในไฟล์ Palindrome.php

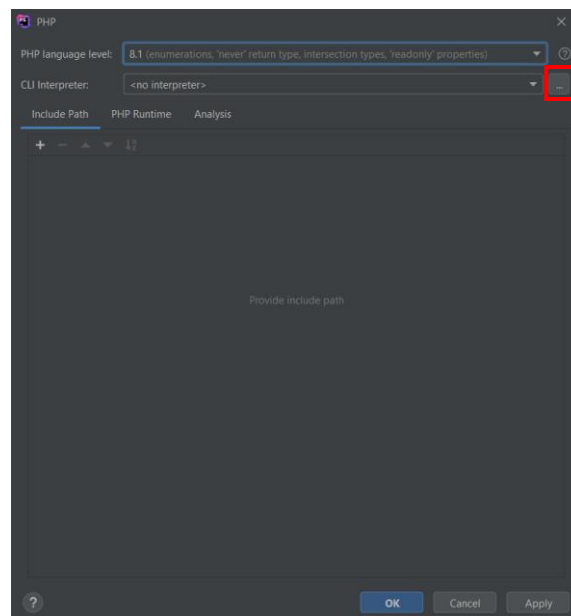


```
1 <?php
2
3 namespace App;
4
5 class Palindrome
6 {
7     public function isPalindrome(string $word): bool
8     {
9         return strtolower($word) === strrev(strtolower($word));
10    }
11 }
12
```

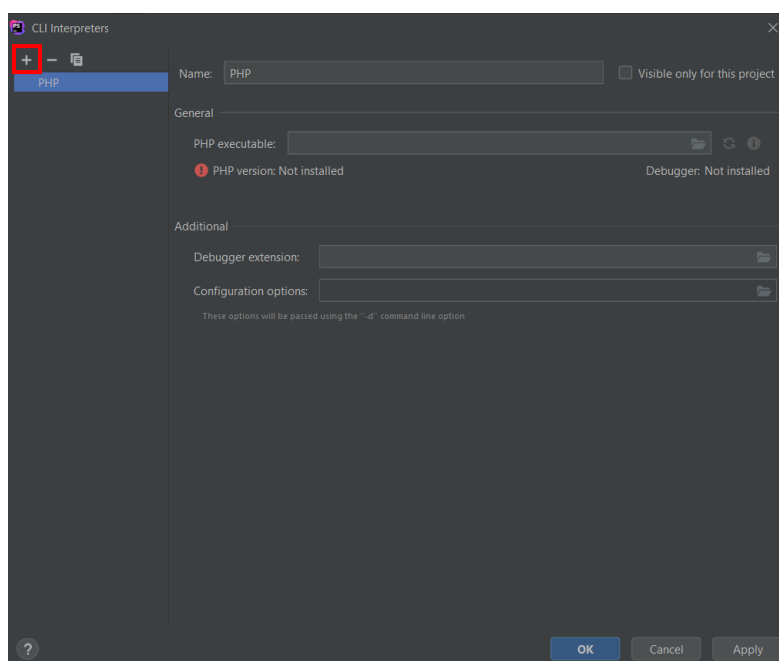
เมื่อกดรันจะขึ้นหน้าจอ Edit Configuration ให้ทำการเลือก Fix



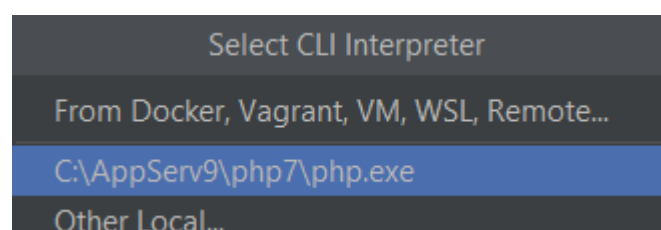
ในหน้า PHP ตรง CLI Interpreter ให้เลือก ...



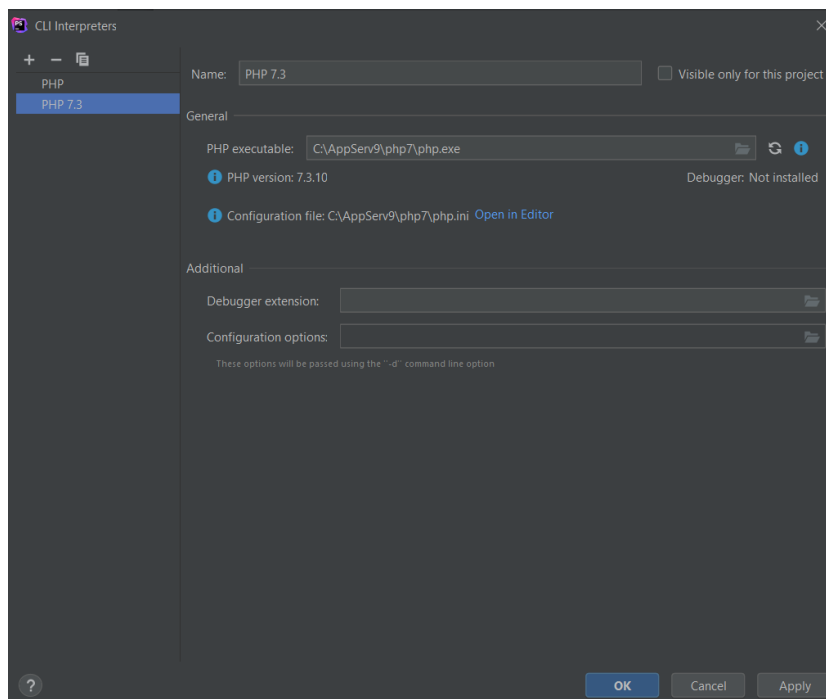
พอได้หน้า CLI interpreters ให้ทำการคลิก + ที่มุมด้านซ้าย



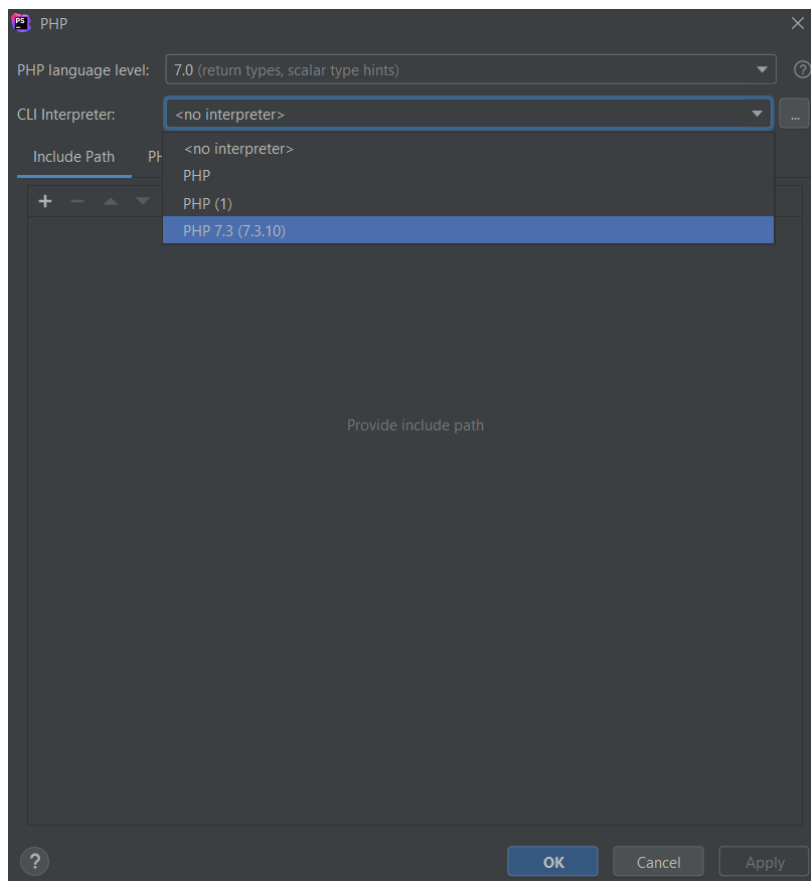
จะมีหน้าต่างขึ้นมาให้เลือก C:\AppServ9\php7\php.exe



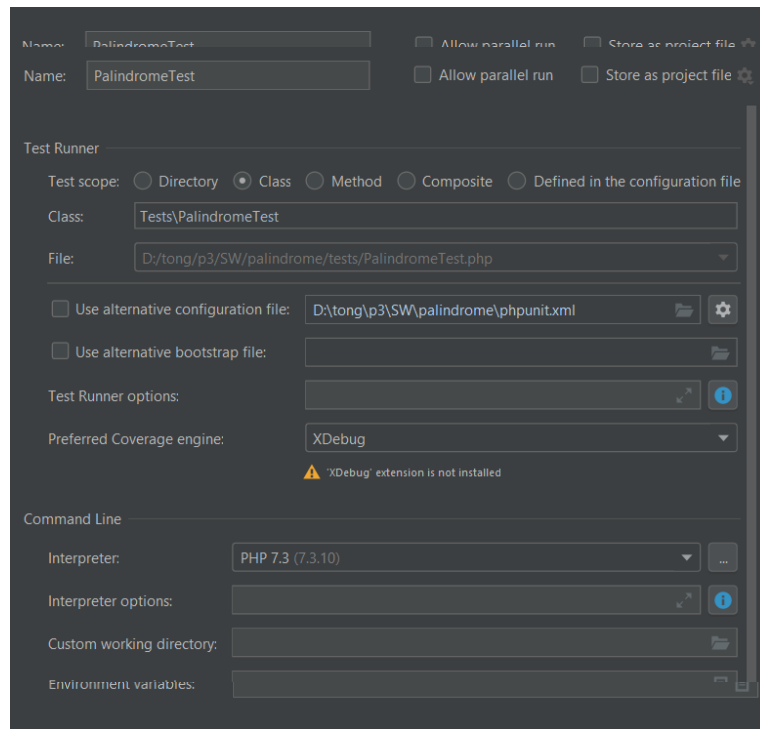
ให้ทำการเลือก PHP 7.3 แล้วคลิกปุ่ม Apply และปุ่ม OK



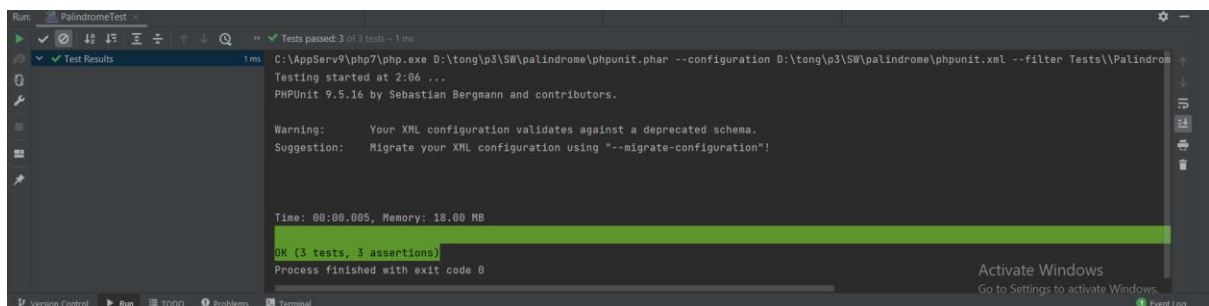
เมื่อกลับมาหน้า PHP ให้ทำการเปลี่ยน CLI interpreter เป็น PHP 7.3 แล้วคลิกปุ่ม Apply และปุ่ม OK



พอกลับมาหน้านี้แล้วให้ทำการคลิกปุ่ม RUN



เมื่อรันออกมาจะได้หน้าต่างแบบนี้



โค้ด

- PalindromeTest.php

```
<?php

namespace Tests;

use App\Palindrome;
use PHPUnit\Framework\TestCase;

class PalindromeTest extends TestCase
{
    public function testPalindrome()
    {
        $palindrome = new Palindrome;

        $this->assertInstanceOf(Palindrome::class, $palindrome);
    }

    public function testIsPalindrome()
    {
        $palindrome = new Palindrome;

        $this->assertTrue($palindrome->isPalindrome("radar"));
    }

    public function testIsPalindrome2()
    {
        $palindrome = new Palindrome;

        $this->assertFalse($palindrome->isPalindrome("baby"));
    }
}
```

- Palindrome.php

```
<?php

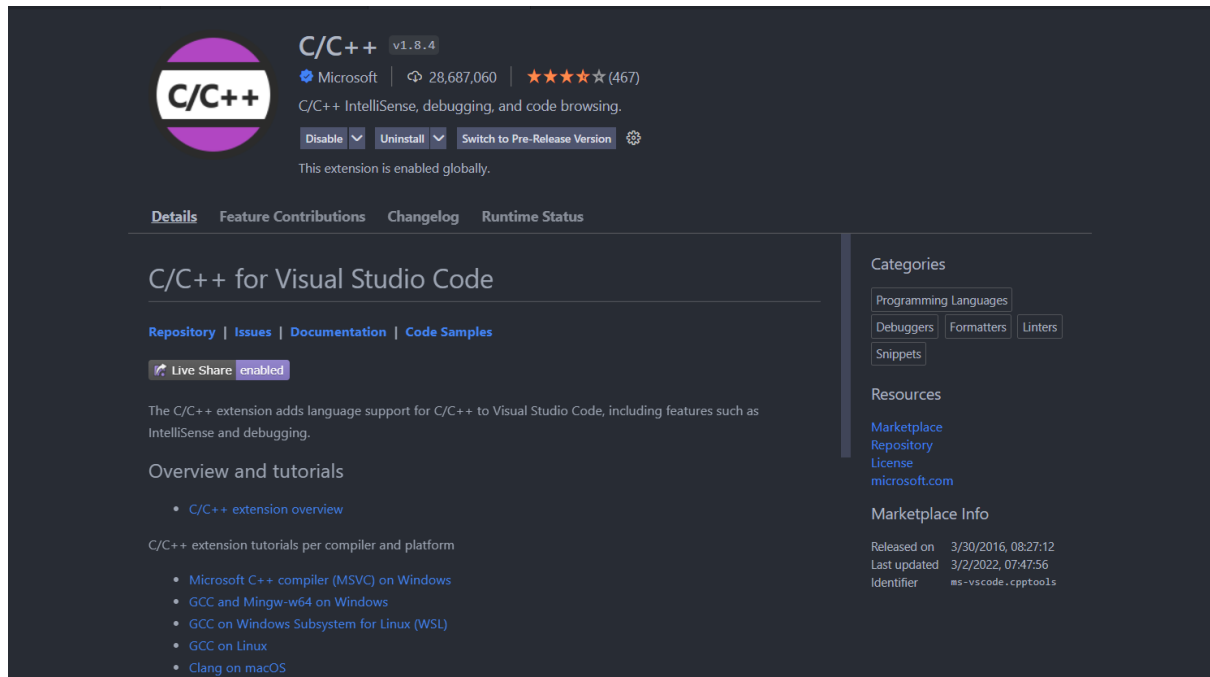
namespace App;

class Palindrome
{
    public function isPalindrome(string $word): bool
    {
        return strtolower($word) === strrev(strtolower($word));
    }
}
```


5.C++

จะทำการรันโปรแกรมผ่าน MSYS2 MinGW x64

เริ่มแรกทำการติดตั้ง Extension



จากนั้นทำการติดตั้ง MSYS2

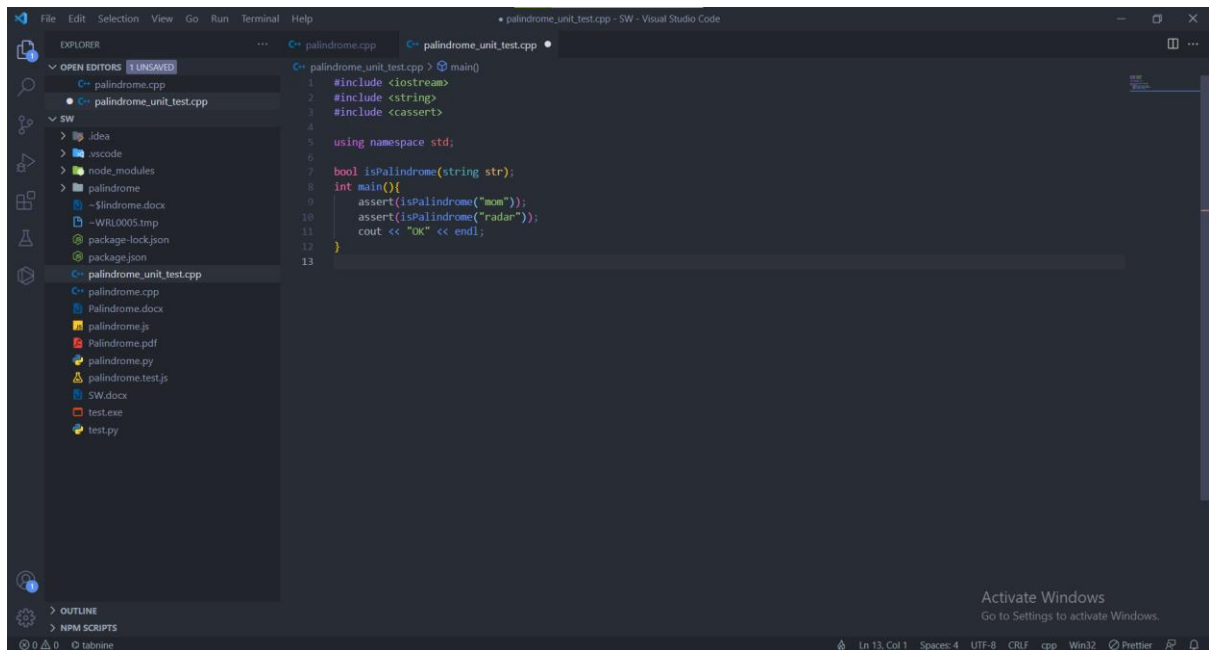
 MSYS2	Rolling	Windows	11.2.0/trunk	Ada, C, C++, Fortran, Obj-C, Obj- C++, OCaml	many
--	---------	---------	--------------	--	------

เมื่อติดตั้ง MSYS2 เสร็จ ให้ทำการสร้างไฟล์ palindrome_unit_test.cpp

แล้วทำการ include

```
1  #include <iostream>
2  #include <string>
3  #include <cassert>
```

ต่อจากนั้นทำการเขียนโค้ด

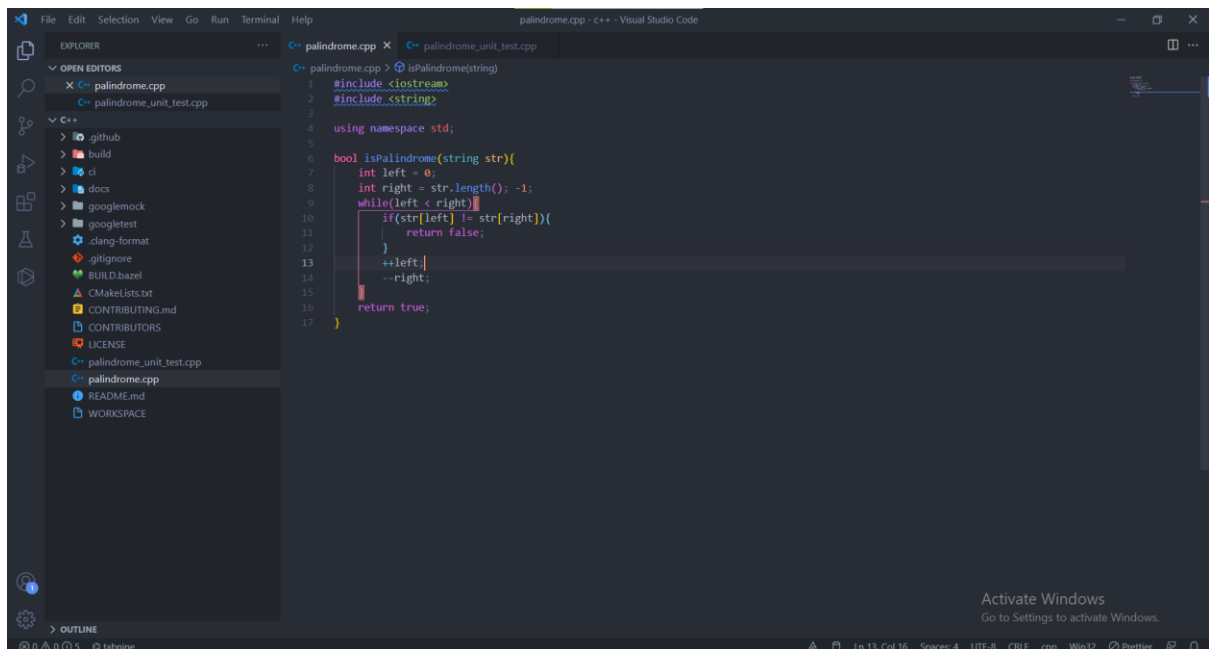


แล้วทำการสร้างไฟล์ palindrome.cpp

ทำการ include

```
1 #include <iostream>
2 #include <string>
```

ต่อจากนั้นทำการเขียนโค้ด

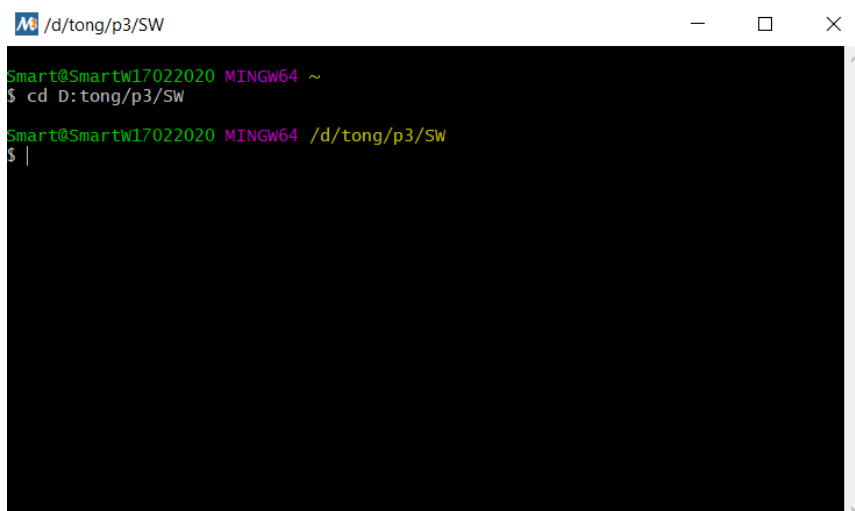


เมื่อเขียนโค้ดเสร็จให้เปิด MSYS2 MinGW x64 ขึ้นมาเพื่อที่จะทำการ Compile ไฟล์



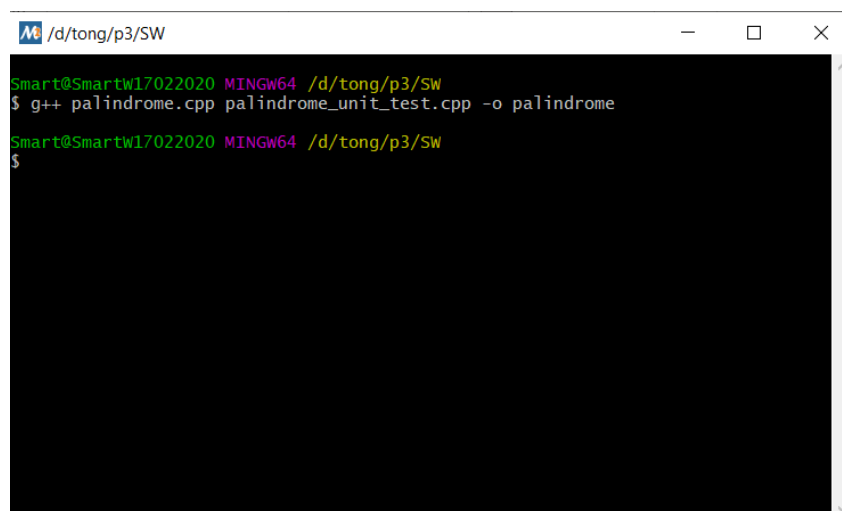
```
Smart@Smartw17022020 MINGW64 ~  
$
```

ทำการ cd เข้าไปที่ตำแหน่งที่ไฟล์ของเราอยู่



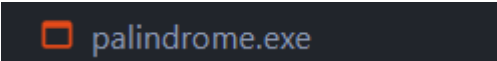
```
Smart@Smartw17022020 MINGW64 ~  
$ cd D:/tong/p3/SW  
Smart@Smartw17022020 MINGW64 /d/tong/p3/SW  
$ |
```

พอเข้ามาแล้วก็ทำการ Compile ด้วยคำสั่ง g++

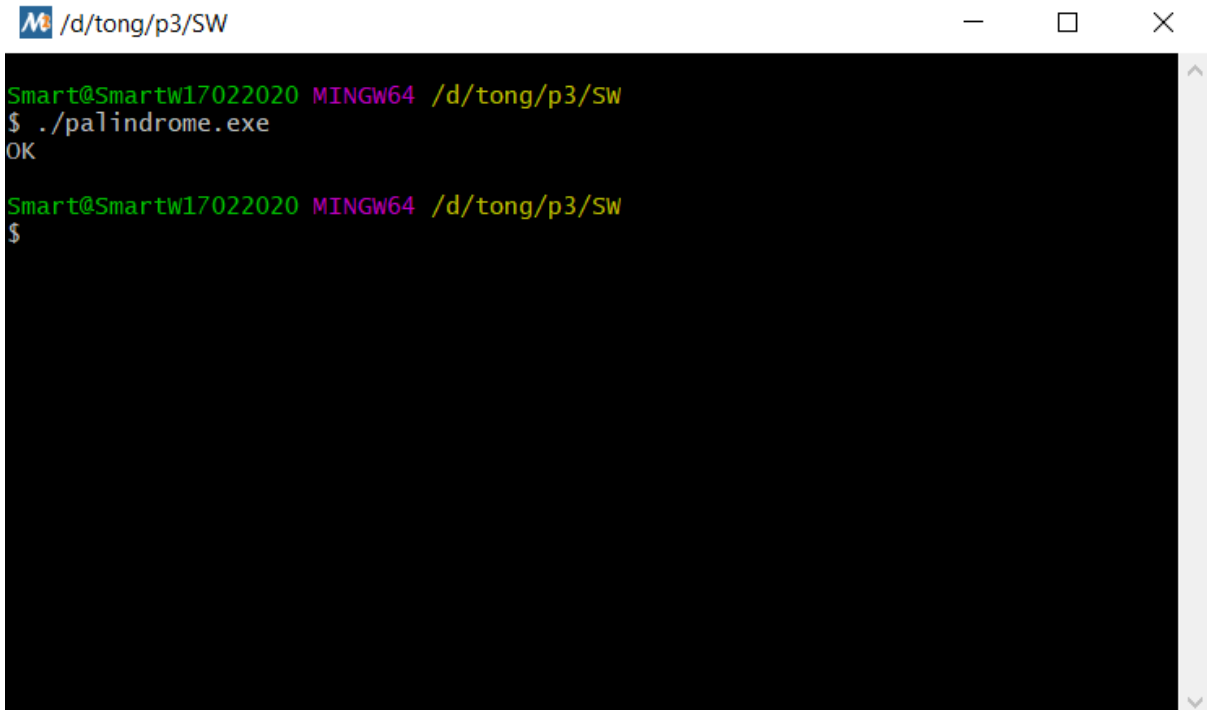


```
Smart@Smartw17022020 MINGW64 /d/tong/p3/SW  
$ g++ palindrome.cpp palindrome_unit_test.cpp -o palindrome  
Smart@Smartw17022020 MINGW64 /d/tong/p3/SW  
$
```

จากนั้นเราก็จะได้ไฟล์ palindrom.exe มา



ลองสั่งรันโปรแกรมดู จะได้ผลลัพธ์ตามภาพ



```
Smart@Smartw17022020 MINGW64 /d/tong/p3/SW
$ ./palindrome.exe
OK
Smart@Smartw17022020 MINGW64 /d/tong/p3/SW
$
```

โค้ด

- palindrome_unit_test.cpp

```
#include <iostream>
#include <string>
#include <cassert>

using namespace std;

bool isPalindrome(string str);
int main(){
    assert(isPalindrome("mom"));
    assert(isPalindrome("radar"));
    cout << "OK" << endl;
}
```

- palindrome.cpp

```
#include <iostream>
#include <string>

using namespace std;

bool isPalindrome(string str){
    int left = 0;
    int right = str.length() - 1;
    while(left < right){
        if(str[left] != str[right]){
            return false;
        }
        ++left;
        --right;
    }
    return true;
}
```