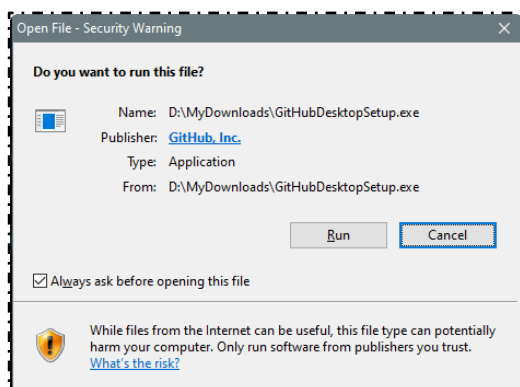


การติดตั้ง GitHub Desktop และการสร้าง Repository

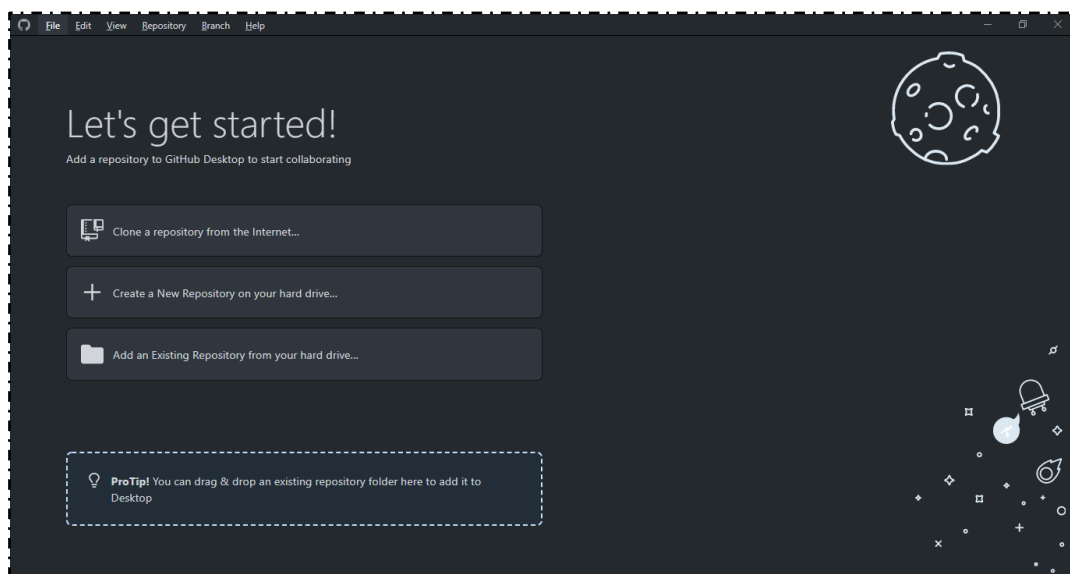
1. เปิด Google Chrome เข้าเว็บไซต์ GitHub Desktop ตามที่อยู่นี้ desktop.github.com



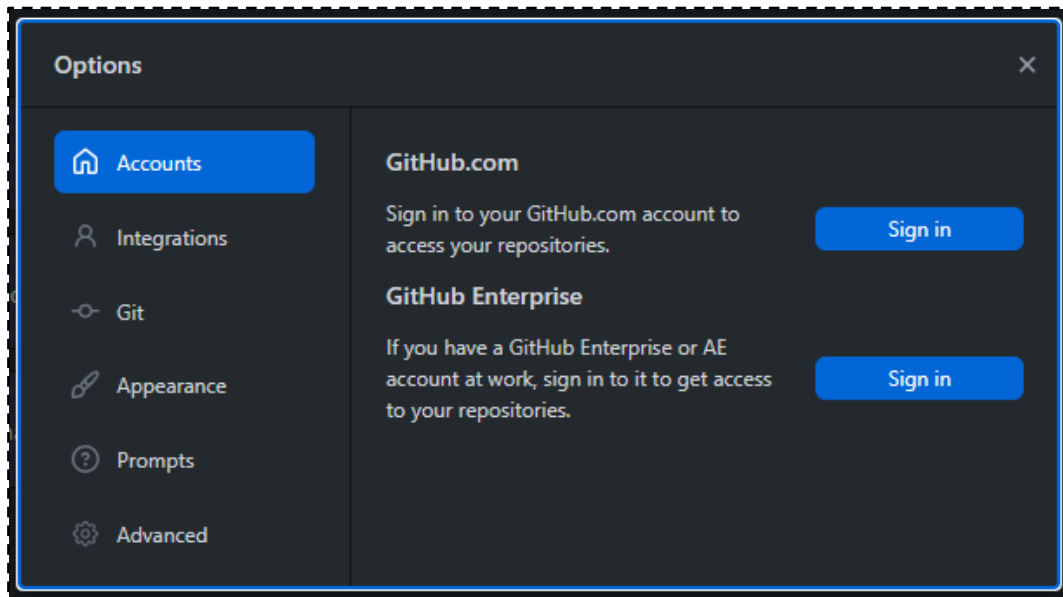
2. เมื่อทำการดาวน์โหลดเรียบร้อยแล้ว ทำการติดตั้ง ดังรูป



3. เมื่อติดตั้งเสร็จเรียบร้อยแล้ว โปรแกรมจะเปิดเองอัตโนมัติ ดังรูป



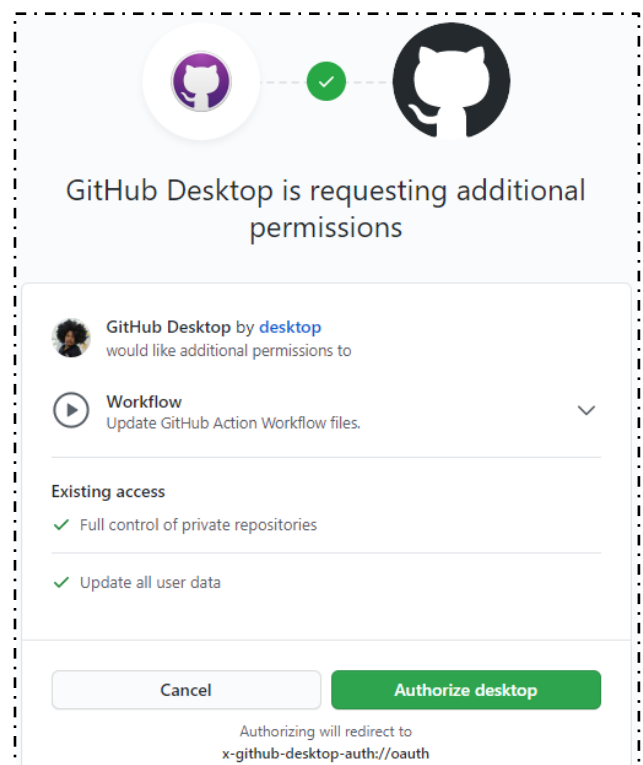
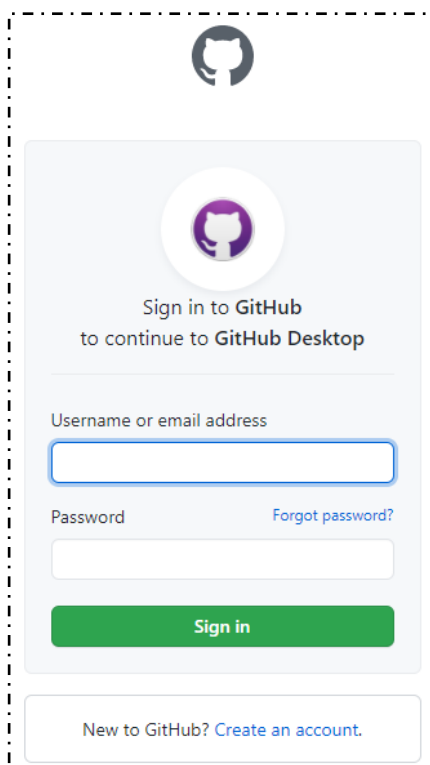
4. เริ่มใช้งานครั้งแรก คลิกเมนู File เลือกเมนูย่อย Options จะมีหน้าต่าง ดังรูป



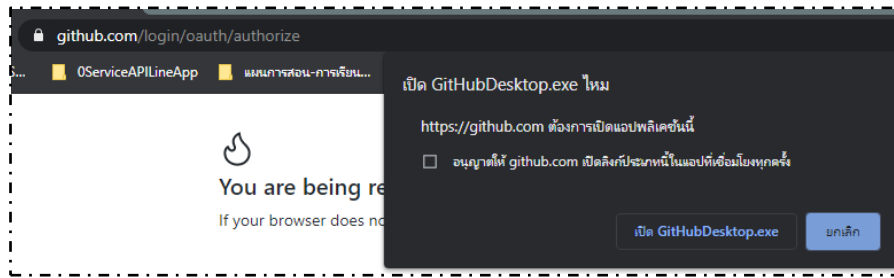
ในหน้าต่างของ Options จะมีเมนูภายใน 6 เมนู คือ

1. Accounts

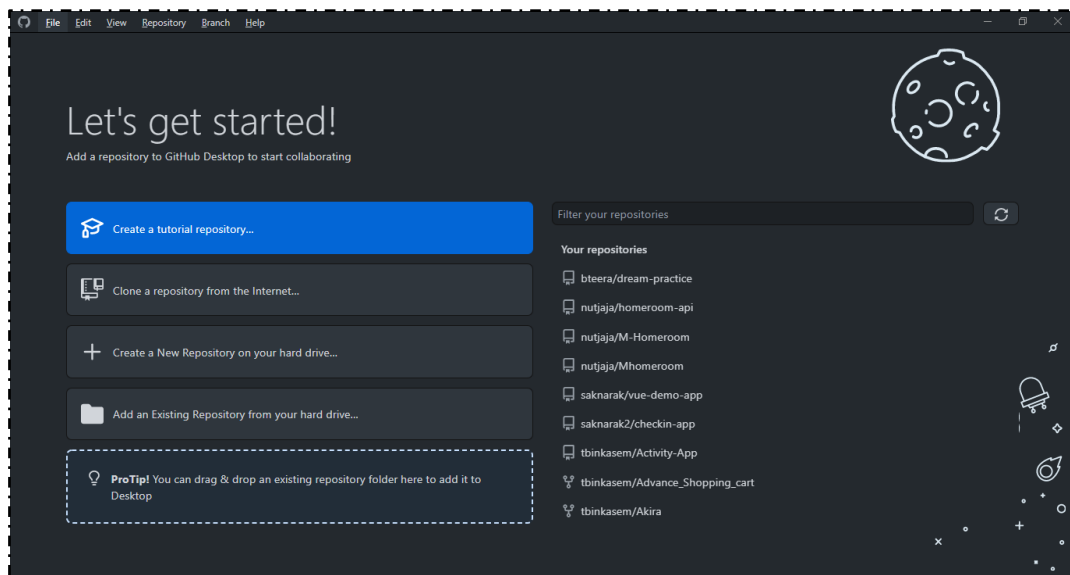
เป็นการลงทะเบียนใช้งาน โดยใช้บัญชี GitHub ที่เราสมัครบนเว็บไซต์ GitHub.com (ฟรี) และการลงทะเบียนใช้งาน โดยใช้บัญชีการใช้งาน GitHub Enterprise (เสียเงิน) ให้คลิกปุ่ม Sign in เพื่อลงทะเบียนใช้งาน จะมีหน้าต่าง ดังรูป เพื่อเปิดการลงทะเบียนผ่าน Web Browser ของเครื่องคอมพิวเตอร์ โดยจะเปิดเว็บไซต์ GitHub.com ให้เรากรอก username/email address ที่เราทำการสมัคร และกรอกรหัสผ่าน เมื่อกรอกเรียบร้อยแล้ว คลิกปุ่ม Sign in



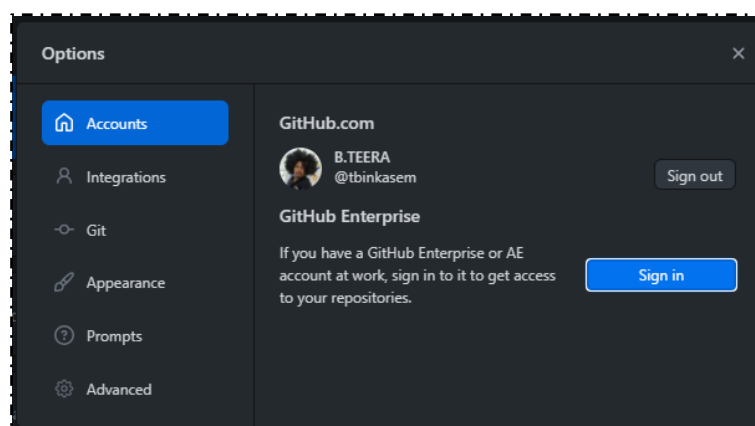
จากนั้นจะมีหน้าเว็บใหม่ให้เราคลิกปุ่ม Authorize desktop เพื่อยินยอม/ยอมรับการใช้งานผ่าน GitHub Desktop จะปรากฏหน้าต่าง เพื่อขออนุญาตให้ github.com เปิดลิงค์ในแอปที่เชื่อมโยง ให้คลิกในช่องสี่เหลี่ยม และกดปุ่ม เปิด GitHubDesktop.exe ดังรูป



หลังจากนั้น จะกลับมาในหน้าต่างโปรแกรม GitHub Desktop อีกครั้ง จะเห็นความเปลี่ยนแปลง คือ จะมีปุ่มสร้าง Repository (Create a tutorial repository) และทางขวามือของโปรแกรมจะแสดงข้อมูล Repository ของเราที่มีใน เว็บไซต์ GitHub.com (หากมี Repo อื่น)



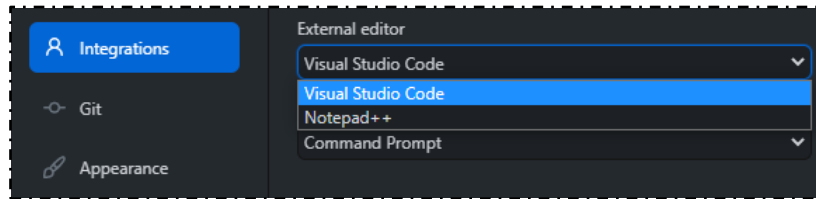
เมื่อเราเปิดเมนู Options อีกครั้งจะเห็นได้ว่า เมนู Accounts จะมีรายละเอียดของบัญชีผู้ใช้ของเรา ดังรูป



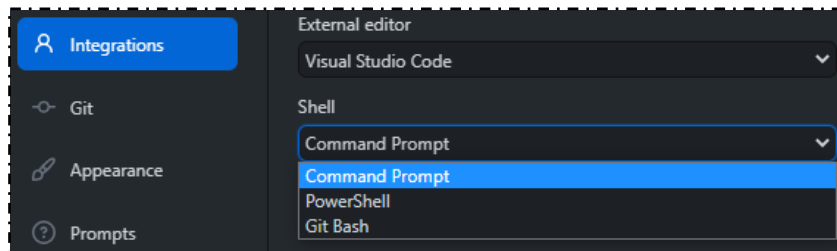
2. Integrations

กำหนดการใช้งานโปรแกรมอื่น ๆ ที่ใช้ร่วมกัน ซึ่งจะมี 2 ส่วนคือ

2.1 External Editor โปรแกรมเขียน Coding หากเครื่องคอมพิวเตอร์ที่เราใช้งานมีโปรแกรมอะไรบ้าง ซึ่งในรูปมีโปรแกรม Visual Studio Code และ Notepad++ ให้เราเลือกได้

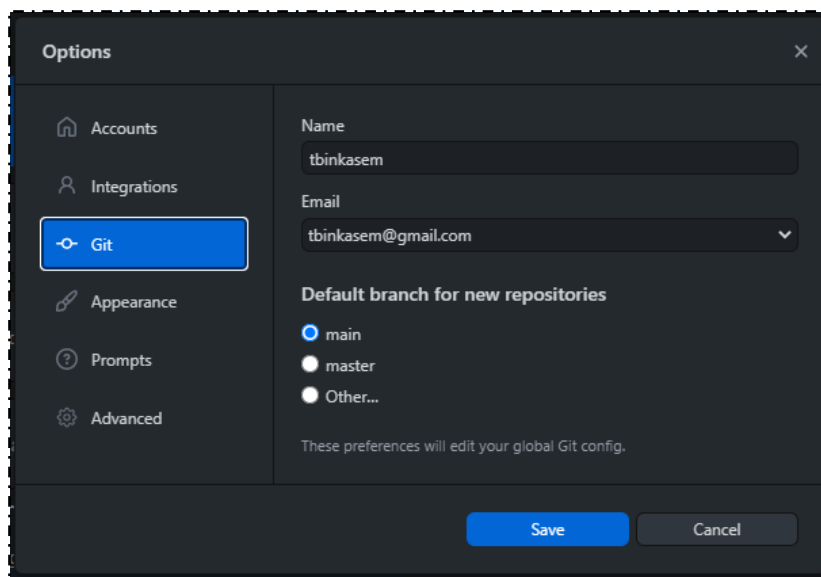


2.2 Shell โปรแกรมใช้คำสั่ง Command Line (CLI) หากเครื่องคอมพิวเตอร์ที่เราใช้งานมีโปรแกรมอะไรบ้าง ซึ่งในรูปมีโปรแกรม Command Prompt, PowerShell และ Git Bash ให้เราเลือกได้



3. Git

เป็นการกำหนด Repository ที่เราจะสร้างใหม่ ให้ใช้ชื่อ Branch หลักอะไร ในหน้าหน้าต่างจะมีให้เลือก 2 ชื่อตามที่กำหนดใน GitHub.com คือ main และ master ส่วนรายการสุดท้ายเราสามารถตั้งชื่อเอง (Other) ได้ โดย Name และ Email มาจากบัญชีที่เราใช้งาน ดังรูป

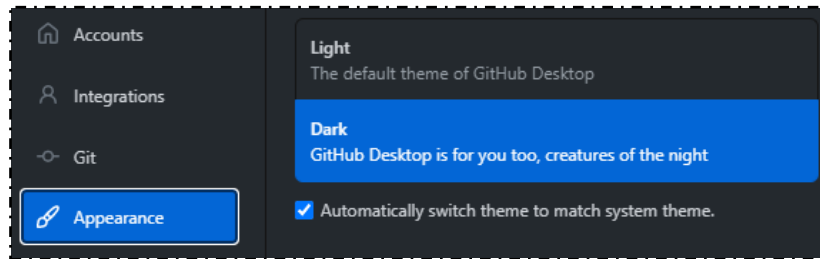


4. Appearance

เป็นการกำหนดรูปแบบสีของโปรแกรม ซึ่งมาให้เลือก 2 โหมด คือ

1. Light แนวโทนสีขาว เป็นโหมดแรกของโปรแกรม
2. Dark แนวโทนมืด เป็นโหมดที่ทำให้สบายตา

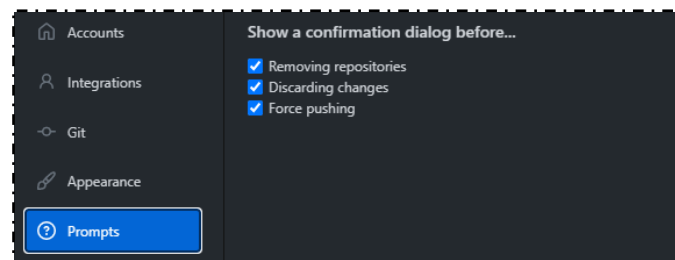
หากเราคลิกเลือกในช่องสไลด์เลือก โปรแกรมจะเลือกโหมดตามลักษณะของระบบปฏิบัติการของเครื่อง



5. Prompts

เป็นส่วนใหญ่ที่กำหนดให้มีการแจ้งเตือน หรือ สอบถามหลังจากการใช้งานคำสั่งต่าง ๆ มี 3 รายการ คือ

- 5.1 Removing repositories กรณีถอน Repository ออกจากโปรแกรม
- 5.2 Discarding changes กรณีมีการยกเลิกการเปลี่ยนของ Repository
- 5.3 Force pushing กรณีมีการ push ข้อมูลทันที



6. Advanced

เป็นส่วนใหญ่ที่กำหนดในการใช้งาน Repository ในกรณีต่าง ๆ

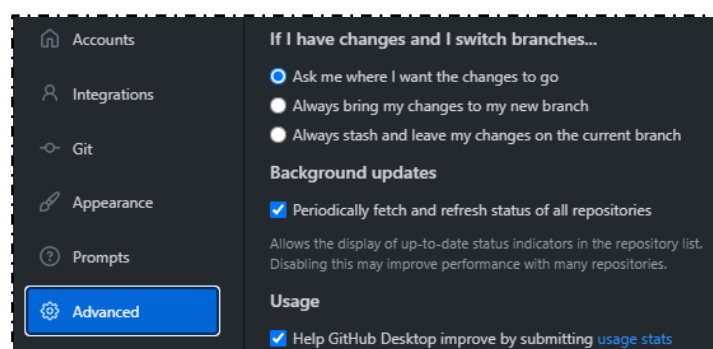
6.1 If I have changes and I switch branches กรณีมีการเปลี่ยน และสลับ branch สามารถเลือกได้ตามต้องการ คือ

- 6.1.1 Ask me where I want the changes to go
ถามทุกครั้งที่มีการเปลี่ยน branch
- 6.1.2 Always bring my changes to my new branch
ถามเฉพาะที่มีการใช้ branch ใหม่
- 6.1.3 Always stash and leave my changes on the current branch
ถามเฉพาะกรณีที่ซ่อนและออกจาก branch หลัก

6.2 Background updates

กำหนดการ fetch และ refresh สถานะของ Repository ทั้งหมดเป็นระยะ ๆ

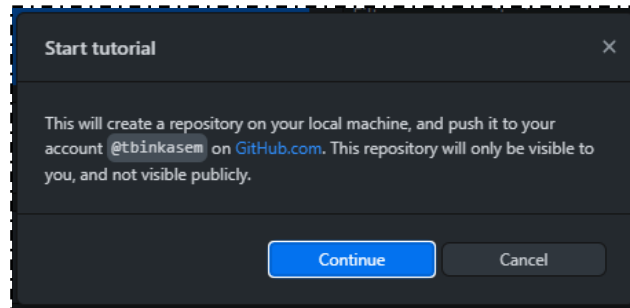
6.3 Usage กำหนดใช้ Help ของ GitHub Desktop เพื่อให้ความช่วยเหลือในการใช้งาน



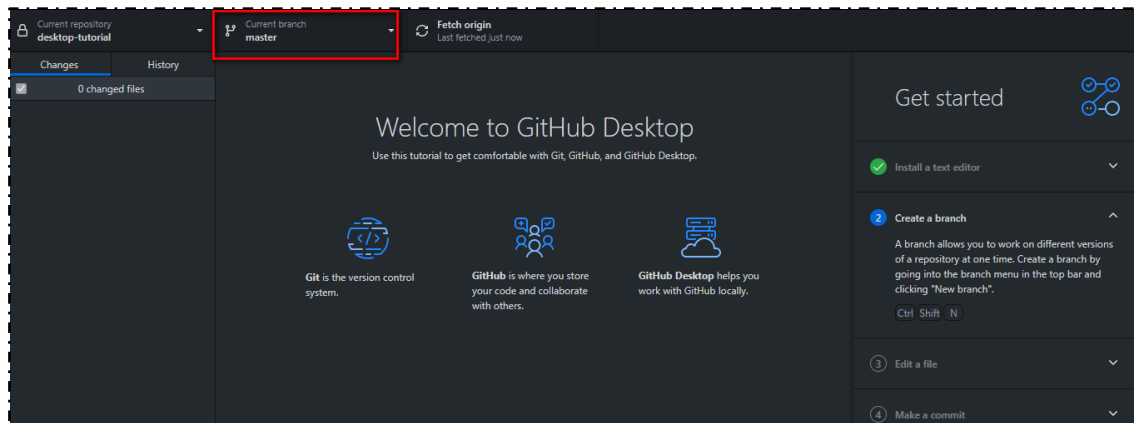
5. การสร้าง Repository

5.1 Create a tutorial repository เป็นการสอนการสร้าง Repository ของตนเองบนเครื่องคอมพิวเตอร์ที่ใช้งานอยู่ปัจจุบัน และจะทำการ push รายการต่าง ๆ ไปยังบัญชี GitHub ที่ได้ลงใช้งาน โดย Repository เราสามารถมองเห็นได้ แต่ไม่สามารถมองเป็นค่าที่เป็นสาธารณะ ขั้นตอนในการสร้าง ดังนี้

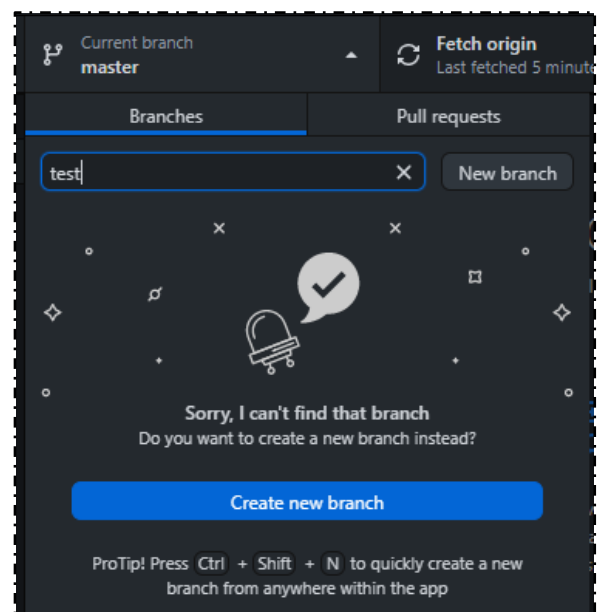
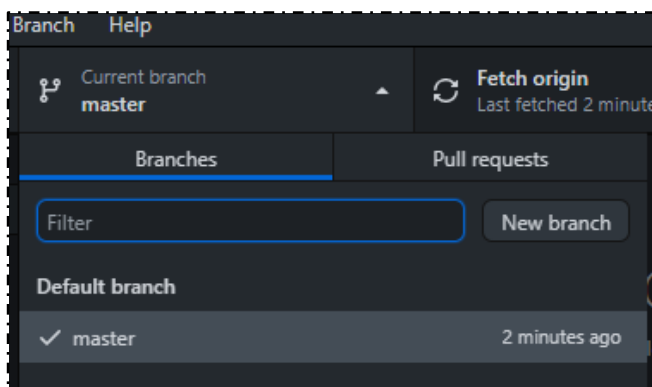
5.1.1 คลิกปุ่ม Continue เพื่อดำเนินการต่อไป

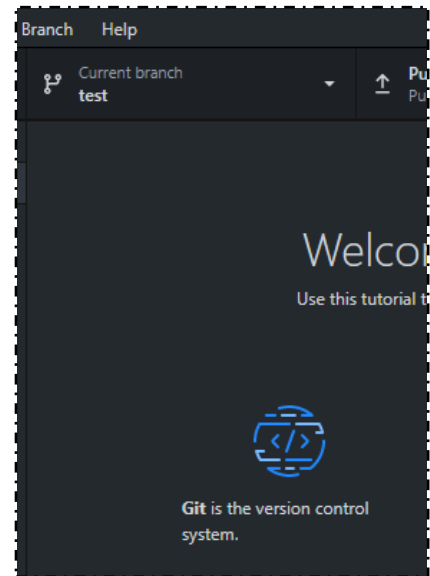
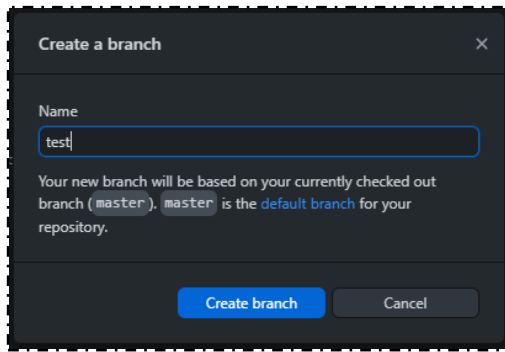


5.1.2 ทำการคลิก Current branch ดังรูป

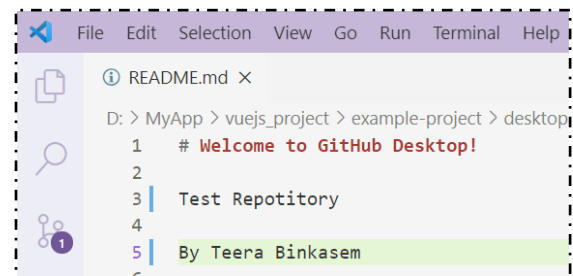
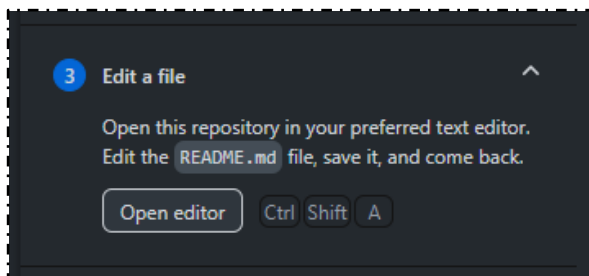


5.1.3 ทำการสร้างชื่อ branch ใหม่ โดยการพิมพ์ชื่อที่ต้องการตั้ง เช่น test แล้วกดปุ่ม Create new branch

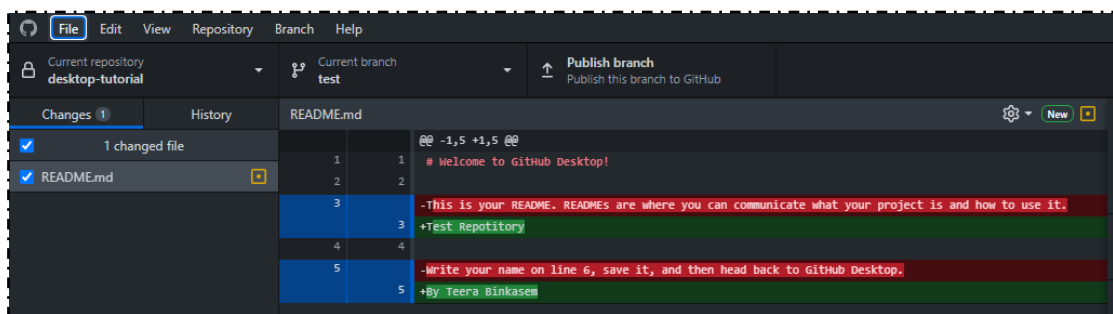




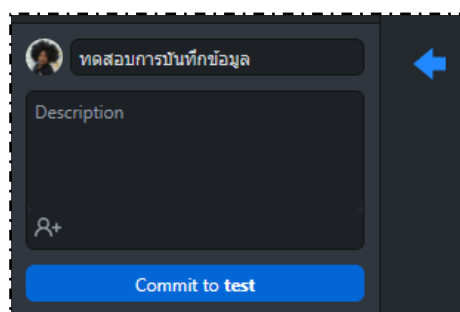
5.1.4 ทำการแก้ไขไฟล์ README.md โดยการเปิดไฟล์ในโปรแกรม Visual Studio Code ที่ทำการตั้งค่าไว้ก่อนหน้านี้ โดยการกดปุ่ม Open editor ทำการแก้ไขภายในไฟล์ แล้วบันทึก



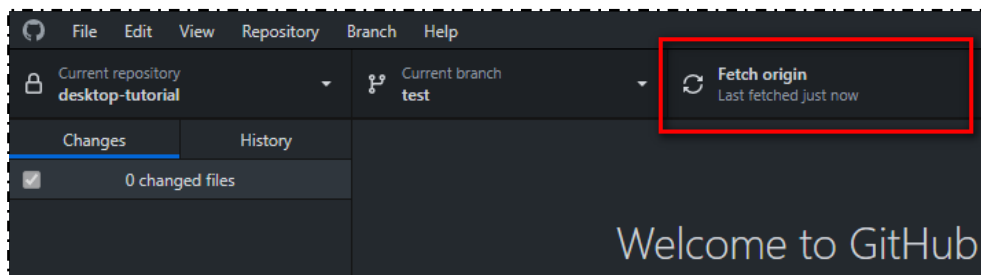
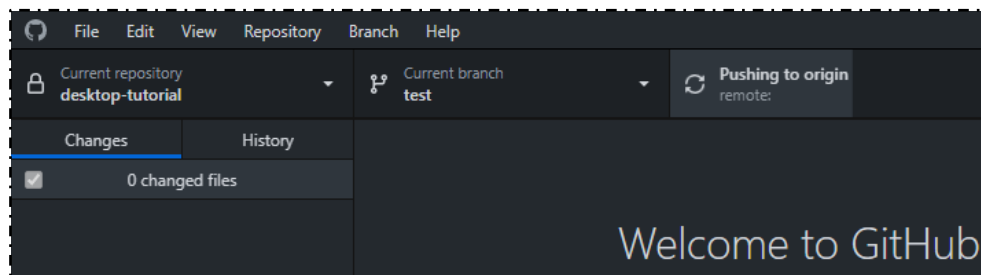
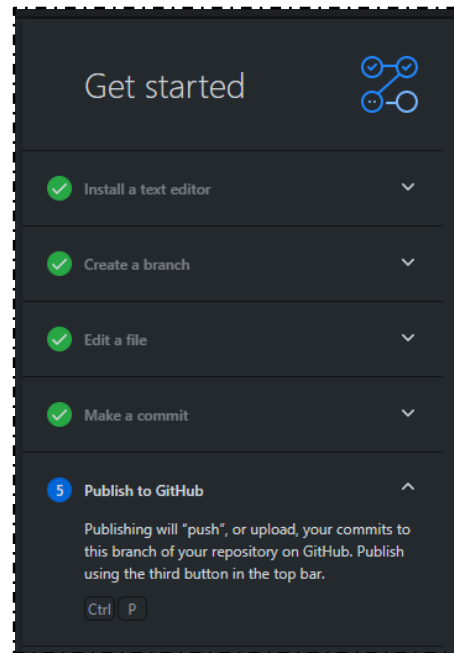
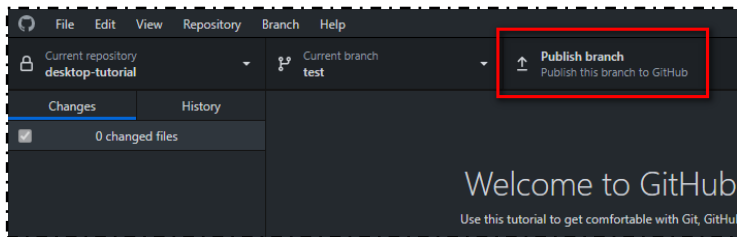
5.1.5 เมื่อเรากลับมาดูใน GitHub Desktop จะเห็นว่าหน้าตาของโปรแกรมเปลี่ยนไป โดยจะทำการแสดงรายการคำสั่งภายในไฟล์ README.md ว่าเปลี่ยนแปลงอะไรไปบ้าง



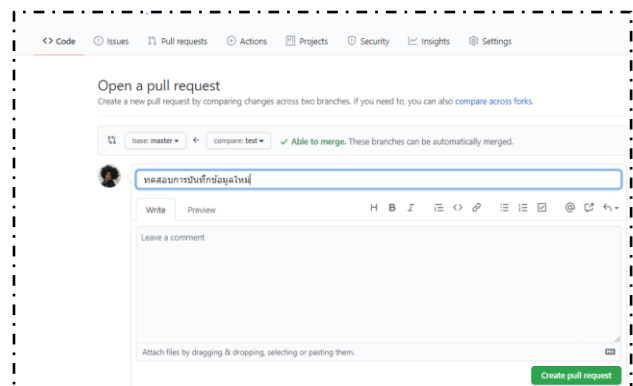
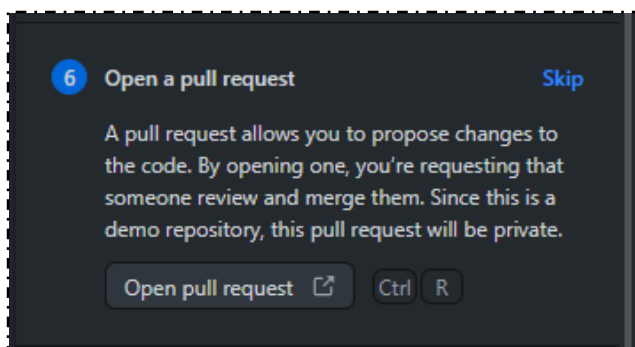
5.1.6 หลังจากนั้นทำการ commit ข้อมูล โดยการพิมพ์ในช่องบน Summary (บังคับข้อมูล) ส่วนช่องล่าง (Description) ที่เป็นช่องใหญ่ (พิมพ์ก็ได้/ไม่ก็ได้) เสร็จแล้วให้กดปุ่ม Commit to test



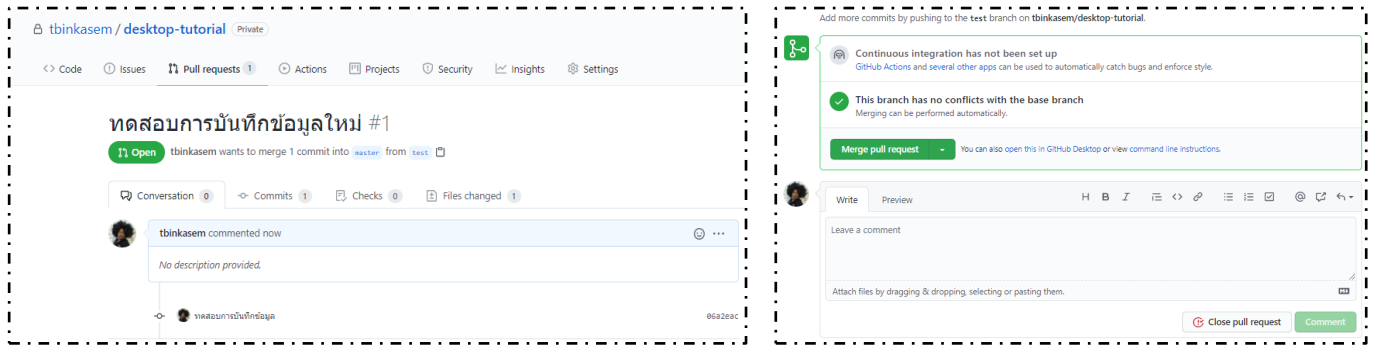
5.1.7 ทำการ push ข้อมูลที่มีการเปลี่ยนแปลง ไปยัง Repository ดังรูป



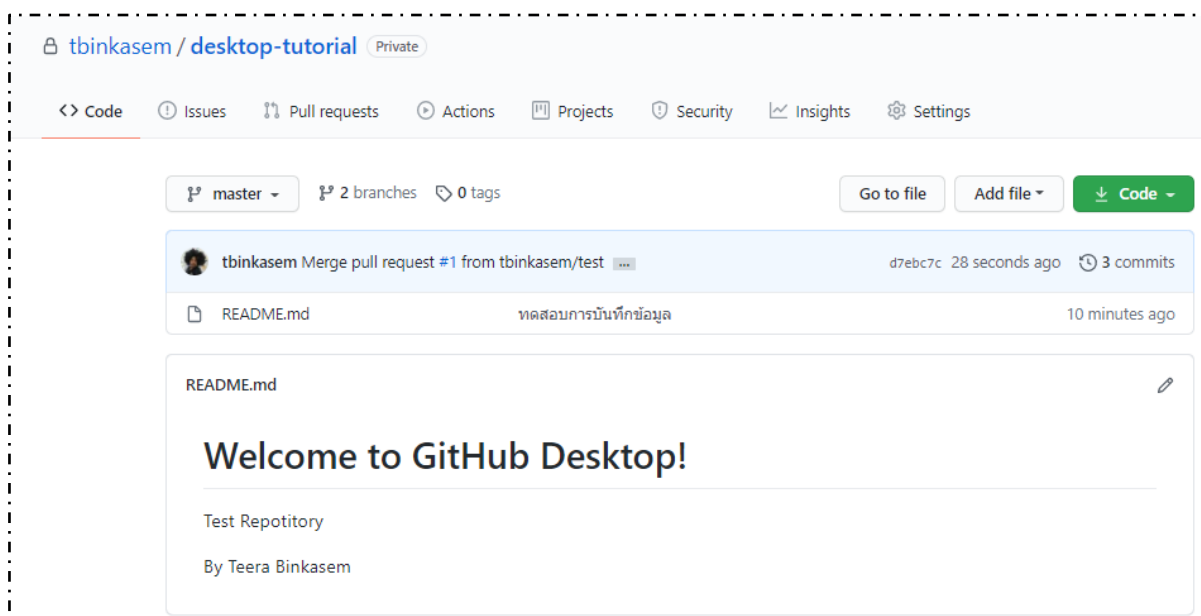
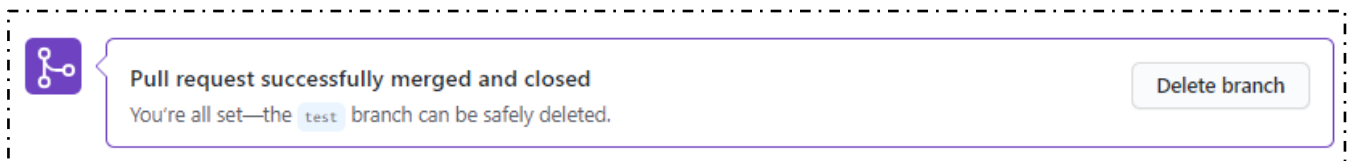
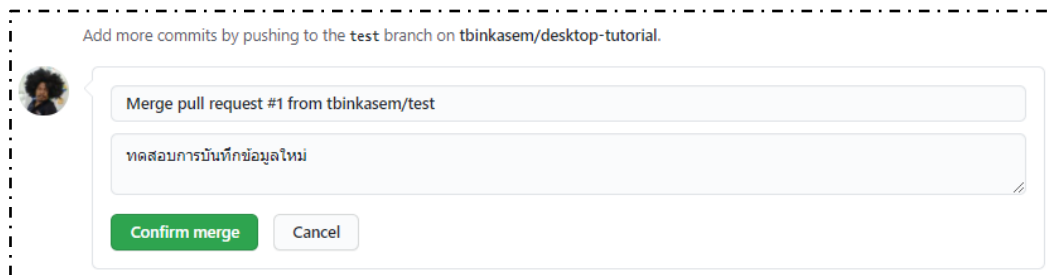
5.1.8 ทำการ pull request เป็นการดึงข้อมูลล่าสุดจาก Repository บนเว็บ GitHub.com มาไว้ในเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ โดยการเปิด GitHub บน Web Browser ดังรูป



5.1.9 ทำการแก้ไขข้อความบางส่วน แล้วกดปุ่ม Create pull request



5.1.10 ทำการรวมการแก้ไข โดยการกดปุ่ม Merge pull request



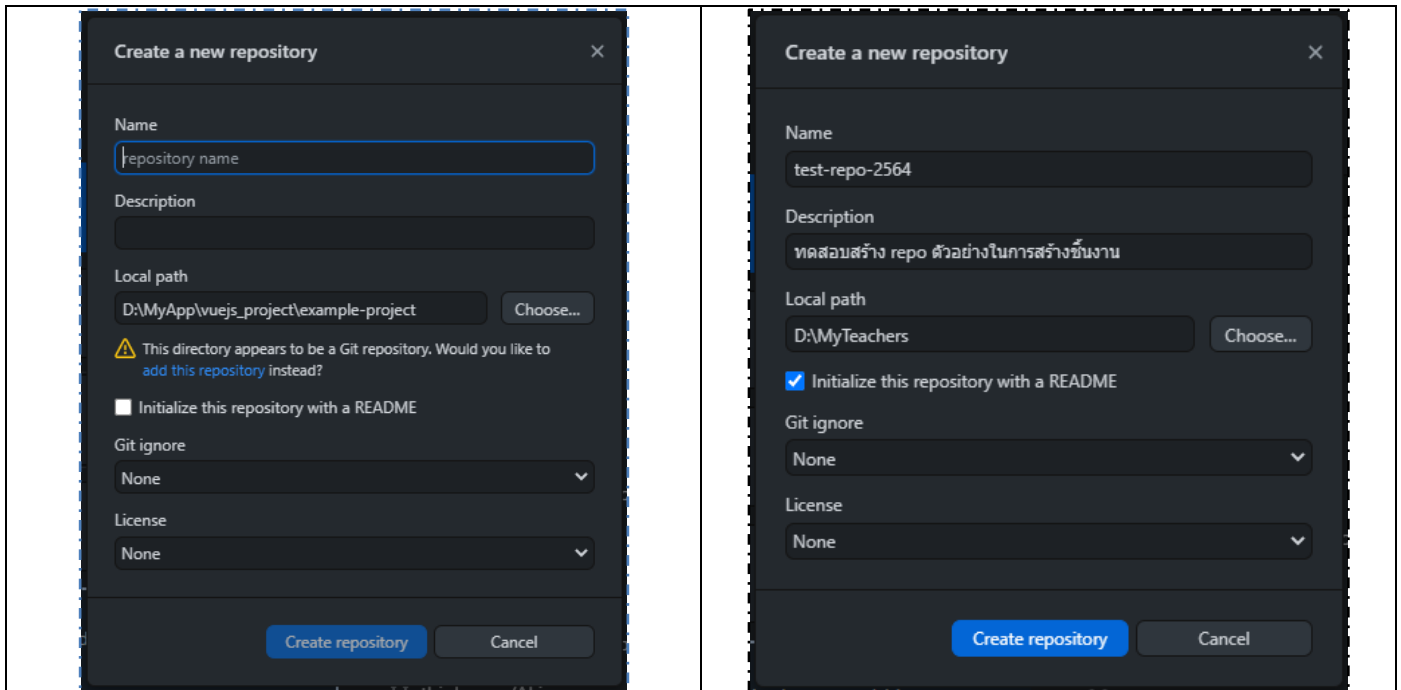
5.2 Create a New Repository on your hard drive เป็นการสร้าง Repository ใหม่ ไปยังเว็บไซต์

GitHub.com พร้อมทั้งสร้างโฟลเดอร์ที่มีชื่อเหมือนกับ Repository ที่สร้างลงบนฮาร์ดดิสก์ของคอมพิวเตอร์ที่ใช้งานอยู่ โดยมีขั้นตอนในการสร้าง ดังนี้

+ Create a New Repository on your hard drive...

5.2.1 ทำการคลิกกรอบข้อความ

หลังจากนั้นจะมีหน้าต่างโปรแกรมขึ้นมา ให้เราทำการกรอกข้อมูล ดังรูป



Name

กำหนดชื่อ Repository ที่ต้องการสร้าง ห้ามซ้ำกับ Repository ที่เคยสร้างแล้ว ข้อกำหนดของชื่อ ต้องเป็นตัวอักษรอังกฤษ, ตัวเลข และเครื่องหมายลบ (-) เช่น test-php-2564 เป็นต้น

Description

คำอธิบายเกี่ยว Repository ที่เราสร้าง สามารถใช้ตัวอักษรภาษาอะไรก็ได้

Local path

ตำแหน่งที่ตั้งของ Repository ที่จะสร้างลงในเครื่องคอมพิวเตอร์ที่ต้องการ เช่น Drive C: หรือ Drive D: หรือสร้างลงในโฟลเดอร์อื่น ๆ ได้

Initialize this repository with a README

ทำการสร้างไฟล์ README.md ลงใน Repository และ โฟลเดอร์ของ Repository ที่อยู่ในเครื่องคอมพิวเตอร์ที่ใช้งาน

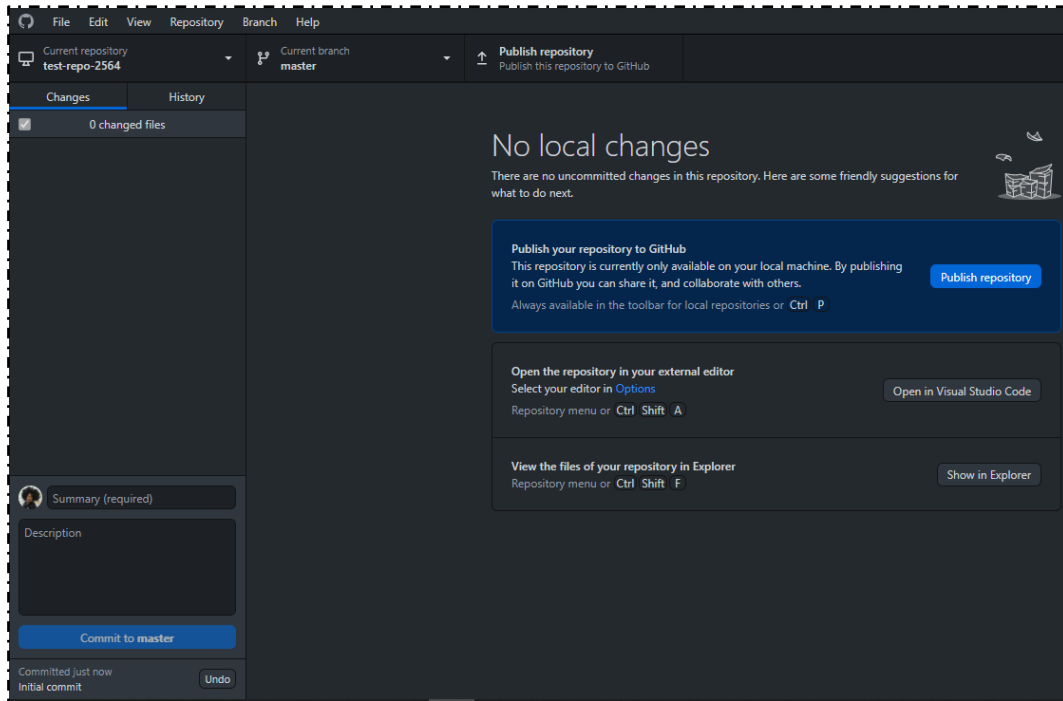
Git ignore

กำหนดให้ยกเว้นไฟล์ของภาษา หรือ นามสกุล หรือ เฟรมเวิร์ก ที่อยู่ในโฟลเดอร์ Repository ด้วย ซึ่งมีหลายภาษาให้เลือก เช่น ภาษา C, cakePHP, dart, Node, Python เป็นต้น หรือถ้ายังไม่กำหนด ให้เลือกเป็น None

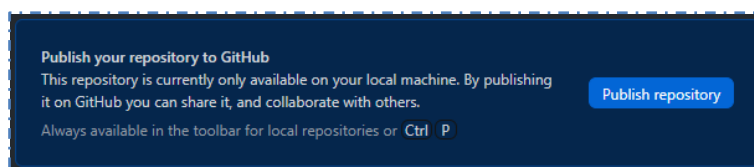
License

กำหนดว่า Repository ที่เราสร้างจะเป็นชิ้นงานในกรอบของลิขสิทธิ์แบบใด เช่น MIT License, GNU GPLv3, Apache License 2.0 เป็นต้น หรือถ้ายังไม่กำหนด ให้เลือกเป็น None

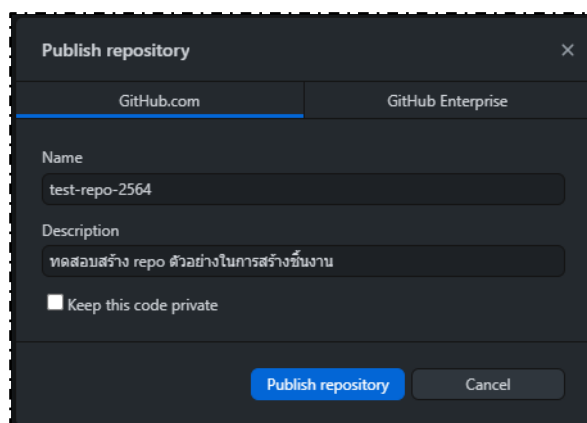
เมื่อกรอกข้อมูลตามที่ต้องการเรียบร้อยแล้ว ให้กดปุ่ม Create repository หน้าตาโปรแกรมจะเปลี่ยนไป โดยจะมีขั้นตอนที่ต้องการกำหนด ดังรูป



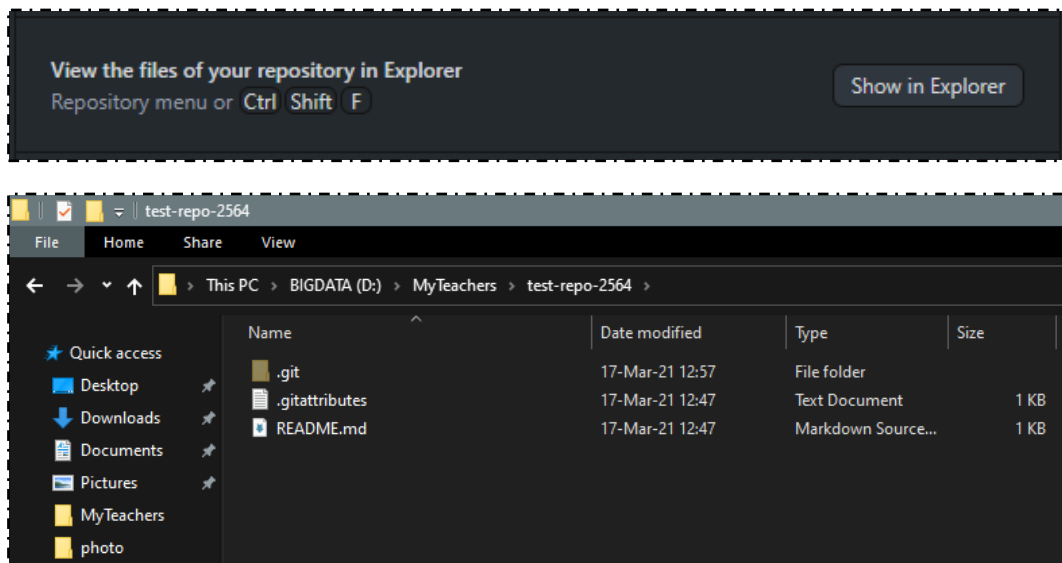
5.2.2 ให้ทำการ Publish repository ลงบนเว็บไซต์ GitHub.com โดยการคลิกปุ่ม Publish repository หรือ ให้ปุ่มคีย์บอร์ด Ctrl + P ก็ได้



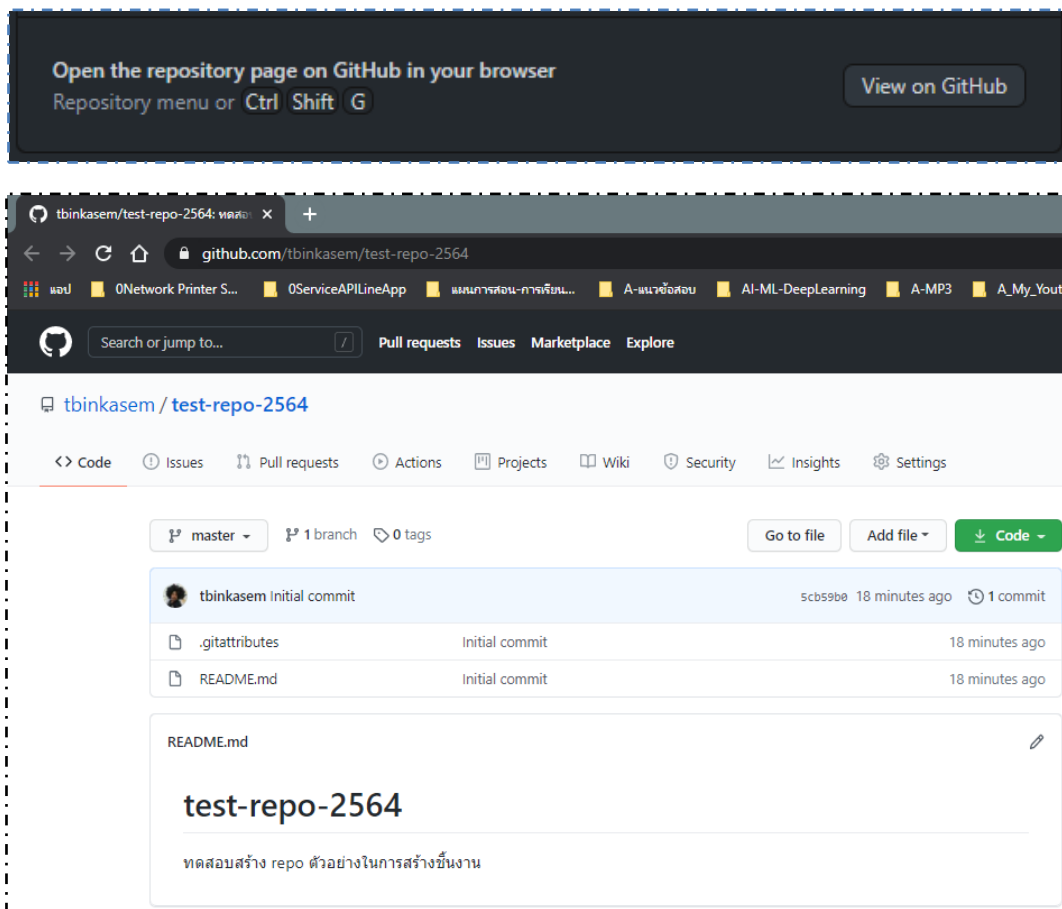
5.2.3 จากนั้นจะมีหน้าต่าง Publish repository ขึ้นมาให้ทำการ คลิกเครื่องหมายในช่องสี่เหลี่ยมออก เพื่อให้ Repository ของที่เราสร้างเป็นแบบ Public ไม่ใช่ Private ทำให้เราสามารถแบ่งปัน ชิ้นงาน หรือ โปรเจคเราได้ ให้แล้วทำการกดปุ่ม Publish repository



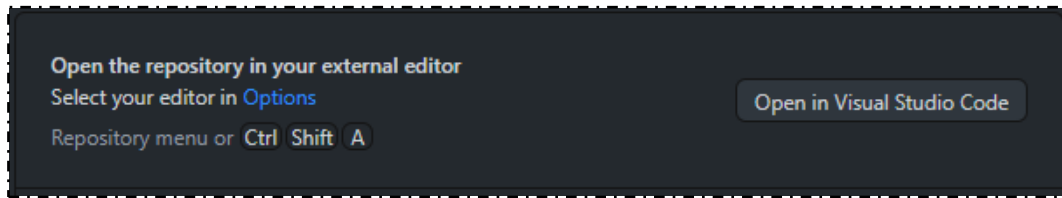
5.2.4 ทำการเปิดโฟลเดอร์ที่เราสร้าง Repository โดยการคลิกปุ่ม Show in Explorer ซึ่งโดยปกติ โฟลเดอร์ .git จะไม่สามารถมองเห็นได้ ในรูปคอมพิวเตอร์ให้กำหนดแสดงไฟล์และโฟลเดอร์ที่ซ่อนไว้ ดังรูป



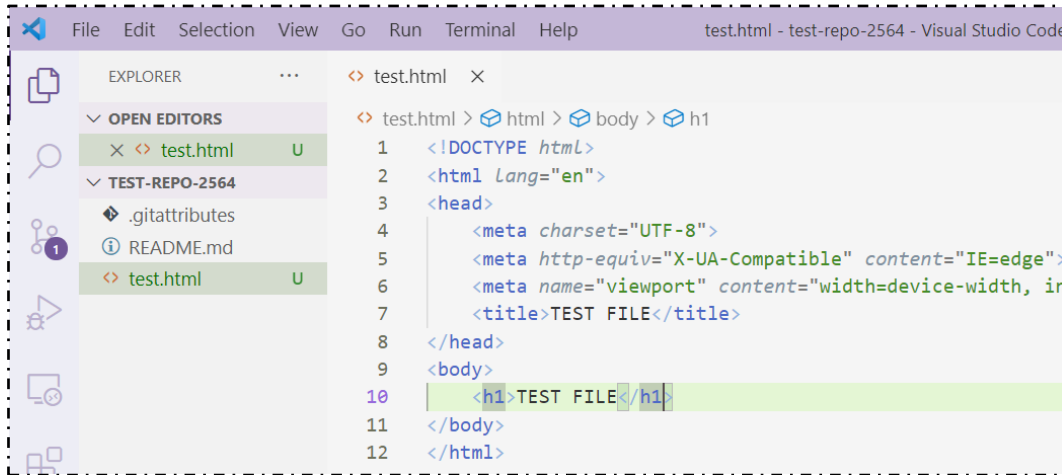
5.2.5 ทำการเปิด Repository ที่เราสร้าง โดยการเปิดโปรแกรม Web Browser โดยการคลิกปุ่ม View on GitHub ดังรูป



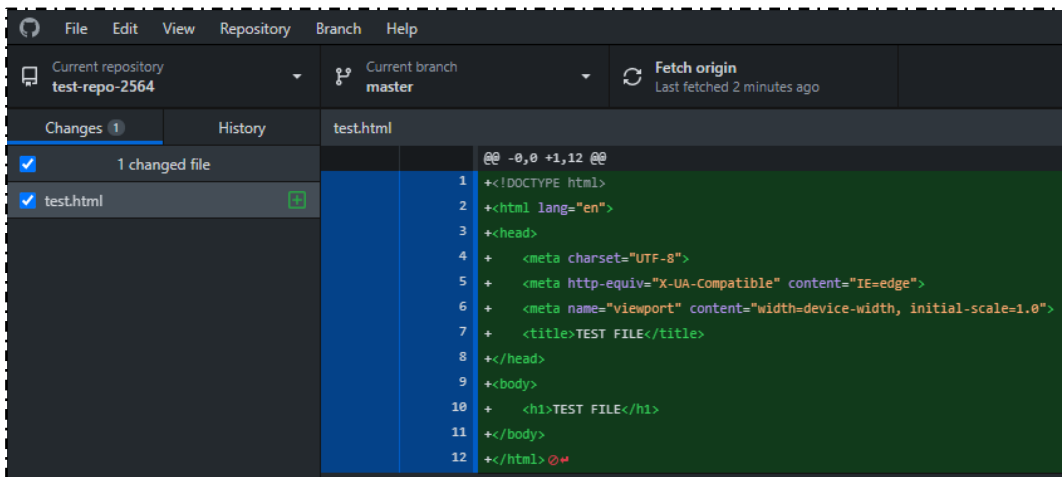
5.2.6 ทำการเปิดโปรแกรม Visual Studio Code เพื่อทำงานในโฟลเดอร์ของ Repository ที่เราสร้าง โดยปกติโฟลเดอร์นี้จะเป็นโฟลเดอร์ชิ้นงาน/โปรเจกต์ด้วย ให้คลิกปุ่ม Open in Visual Studio Code ดังรูป



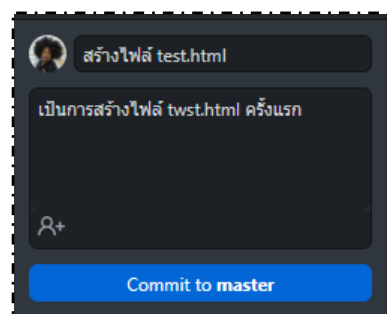
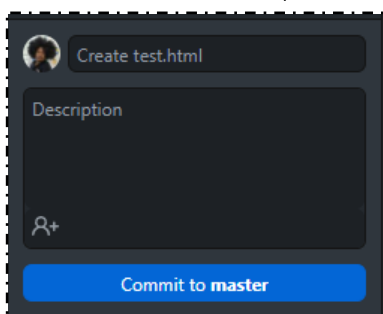
จากนั้นทดลองสร้างไฟล์ชื่อ test.html แล้วทำการเพิ่มแท็ก <h1>TEST FILE </h1> ลงในแท็ก <body> แล้วทำการบันทึกไฟล์ (save) ดังรูป



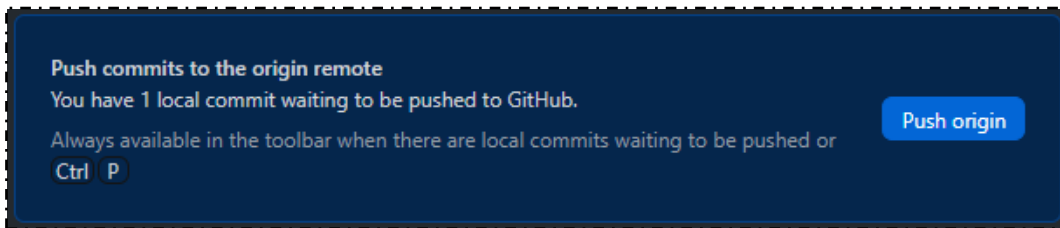
5.2.7 จากนั้นกลับมาดูในโปรแกรม GitHub Desktop จะเห็นว่า โปรแกรมจะแสดงการเปลี่ยนข้อมูลต่าง ๆ ในโฟลเดอร์ Repository ของเรา ดังรูป



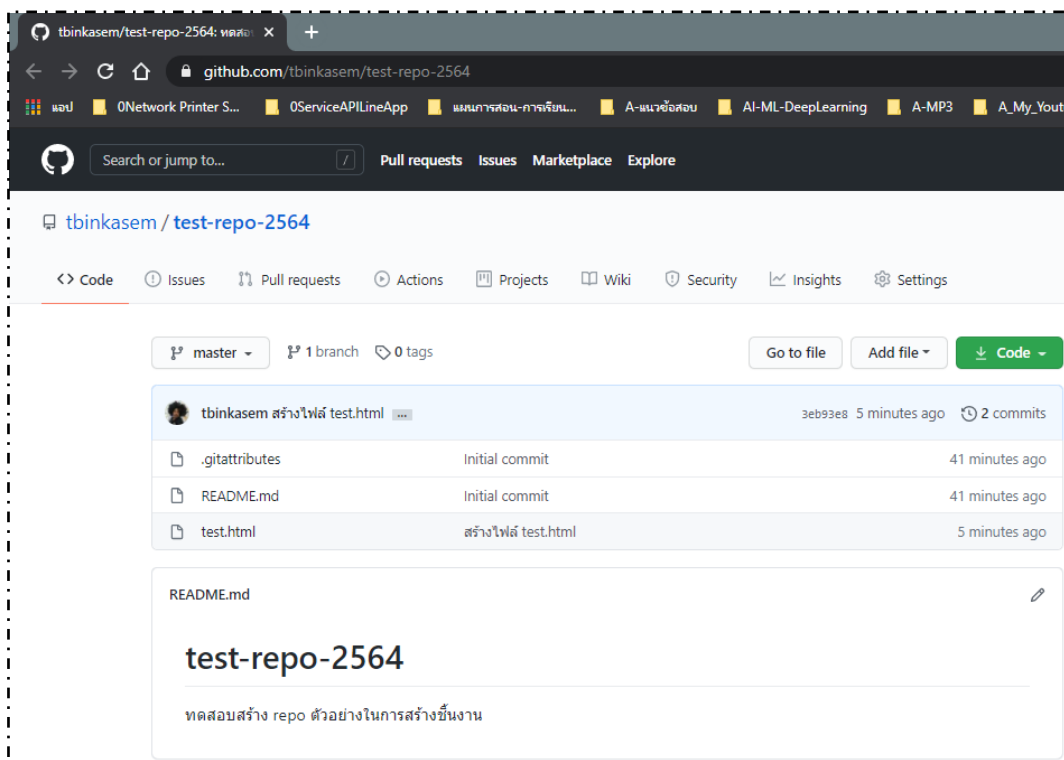
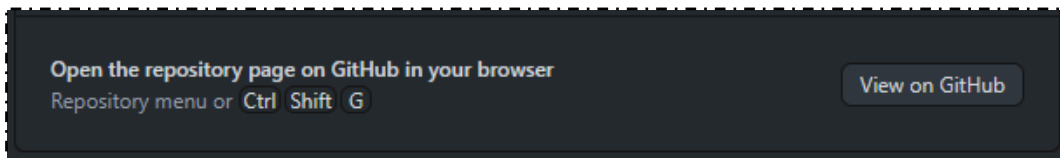
5.2.8 หลังจากนั้น ให้เราทำการ commit ข้อมูลที่เปลี่ยนแปลง โดยการกรอกข้อมูลในกรอบของ Commit และทำการ Commit เสร็จแล้วให้กดปุ่ม Commit to master ดังรูป



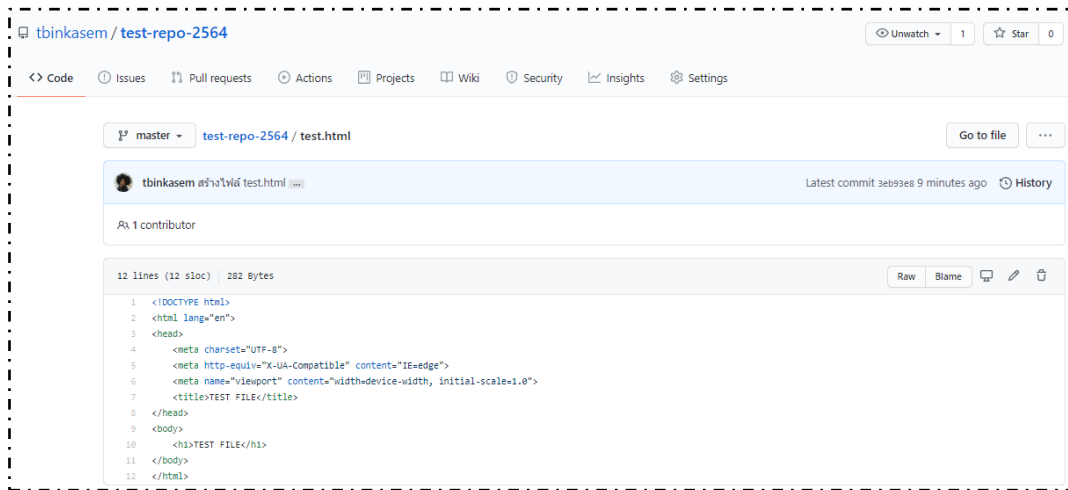
5.2.9 เมื่อทำการ commit เรียบร้อยแล้ว ให้ทำการกดปุ่ม Push Origin ต่อไป ดังรูป



5.2.10 เมื่อกดปุ่ม Push origin แล้วตอนนี้ระบบใน GitHub Desktop จะส่งไฟล์ไปเก็บไว้ใน Repository ของเราในเว็บไซต์ GitHub.com ซึ่งเราสามารถเข้าไปดูได้ในเว็บไซต์ โดยการกดปุ่ม View on GitHub ดังรูป



จะเห็นว่าไฟล์ test.html ไปเก็บไว้แล้ว ข้อมูลในไฟล์เหมือนตัวฉบับที่อยู่ในเครื่องของเราเอง ทดลองคลิกไฟล์ test.html จะเห็นว่ามีแท็กคำสั่งเหมือนกันทุกอย่าง ดังรูป



The screenshot shows the GitHub web interface for a repository named 'test-repo-2564' by user 'tbinkasem'. The file 'test.html' is selected, showing its latest commit. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>TEST FILE</title>
8 </head>
9 <body>
10  <h1>TEST FILE</h1>
11 </body>
12 </html>
```



The screenshot shows the Visual Studio Code editor with the 'test.html' file open. The file explorer on the left shows the file structure. The code in the editor is identical to the one shown in the GitHub screenshot:

```
<> test.html x
<> test.html > html > body > h1
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>TEST FILE</title>
8 </head>
9 <body>
10  <h1>TEST FILE</h1>
11 </body>
12 </html>
```

ดังนั้นในการทำงานต่อไป เราจะสร้างหรือแก้ไข หรือ ลบข้อมูล เมื่อมีการเปลี่ยนแปลงใหม่ทุกครั้ง ระบบของ GitHub จะเก็บข้อมูลไว้ แล้วเราสามารถทำการ commit และ push เป็นระยะ ๆ ต่อไป

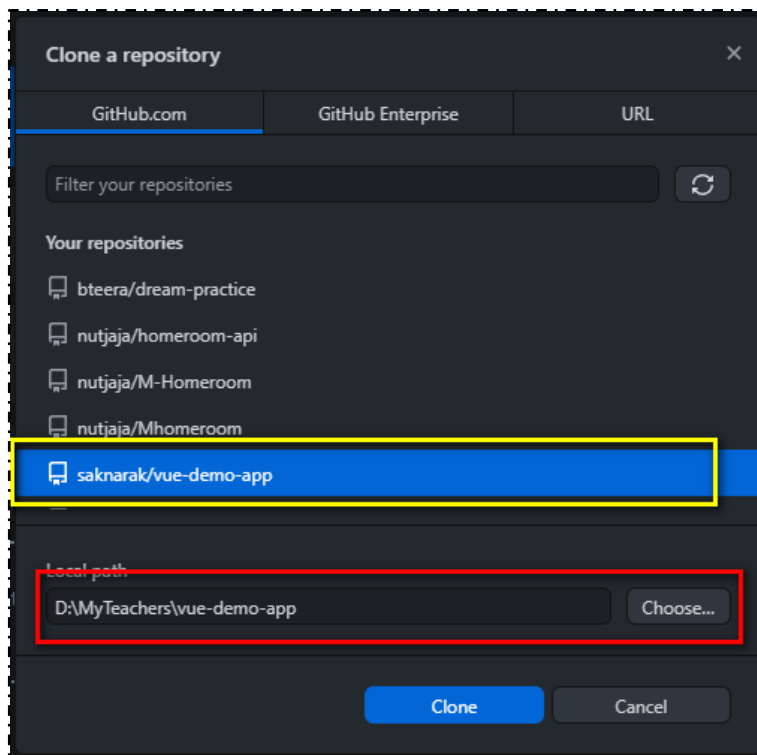
5.3 Clone a repository on the Internet เป็นการคัดลอกข้อมูล/โปรเจกของ Repository ของตนเอง หรือ ของผู้อื่นที่เป็นแบบ Public มาลงในเครื่องคอมพิวเตอร์ของเรา มีขั้นตอนในการสร้าง ดังนี้

5.3.1 คลิกที่ปุ่ม Clone a repository on the Internet จะปรากฏหน้าต่างให้ทำการ Clone (คัดลอก) ให้เราสามารถเลือกวิธีการ clone ซึ่งมีทั้งหมด 3 แบบ คือ

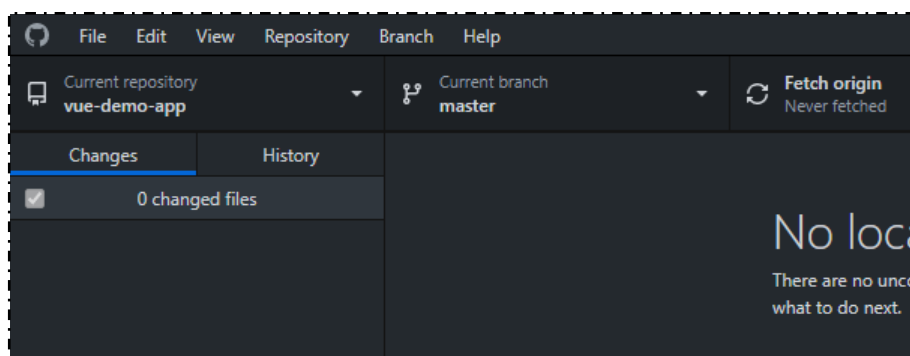
5.3.1.1 แบบใช้ GitHub.com

เป็นการ Clone Repository ของตนเอง หรือ Repository ที่เราทำการ Fork มาเก็บไว้ใน GitHub ของเรา แต่ต้องลงตำแหน่งอื่นในเครื่องคอมพิวเตอร์เดิม โดยมีขั้นตอนในการ Clone ดังนี้

- คลิกที่รายการ Repository ที่ต้องการ Clone จากนั้นโปรแกรมจะสร้างโฟลเดอร์ตามชื่อของ Repository อยู่ภายใต้โฟลเดอร์ที่เราเลือก หรือ ไดรฟ์ที่เราเลือก ดังรูป



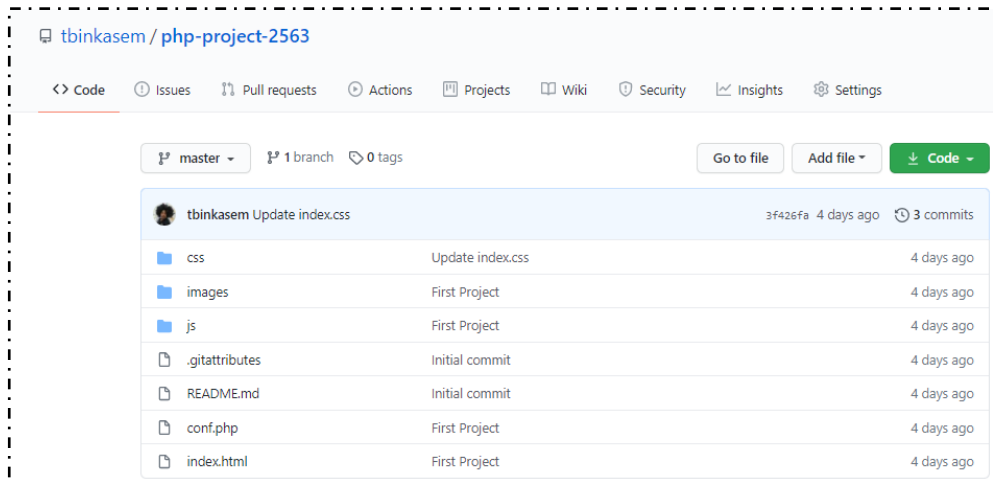
หลังจากนั้นเราก็กดปุ่ม Clone โปรแกรมก็จะสร้างจนเสร็จ จะปรากฏหน้าต่างโปรแกรม ดังรูป ซึ่งการทำงานหลังจากนี้ เหมือนเราใช้งาน Repository ที่เราสร้างขึ้นมาเองใหม่ โดยจะทำให้ Repository ของเราจะมีอยู่ มากกว่า 1 ตำแหน่ง แต่จะเชื่อมโยงกัน หากมีการเปลี่ยนแปลงข้อมูลก็จะเปลี่ยนได้เหมือนเป็นเครื่องเดียวกัน ดังรูป



5.3.1.2 แบบใช้ GitHub Enterprise เป็นการ Clone จาก GitHub Enterprise

5.3.1.1 แบบใช้ URL เป็นการ Clone Repository โดยใช้ URL ของ Repository ที่เราต้องการเว็บไซต์ GitHub.com ของตนเอง หรือ ผู้ใช้งานคนอื่น ๆ ที่เผยแพร่ URL ของ Repository ให้ทำการ Clone ได้ โดยการใช้งานมีขั้นตอนดังนี้

- เปิดเว็บไซต์ GitHub.com ของตนเอง แล้วเลือก Repository ที่ต้องการ Clone มาใช้งานในโฟลเดอร์อื่นในเครื่องเดิม หรือ ในเครื่องอื่น ๆ ดังรูป.

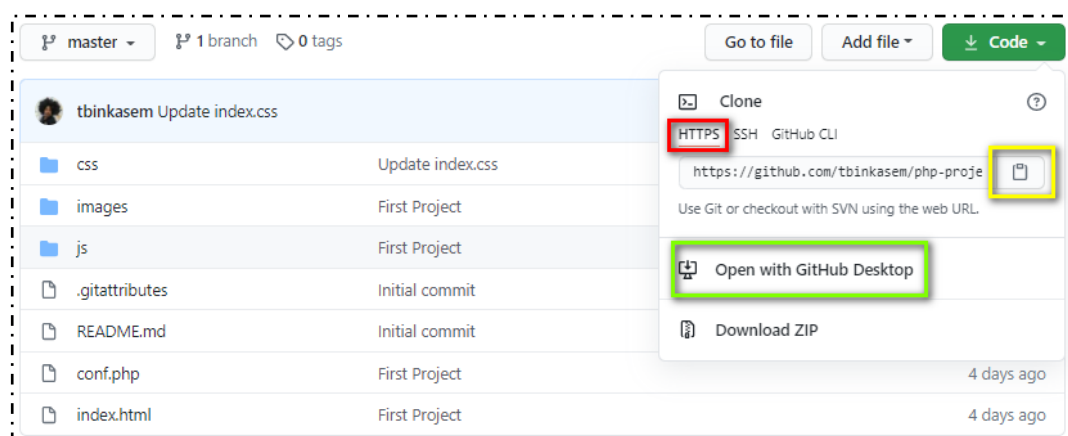


- คลิกที่ปุ่ม Code สีเขียว จะปรากฏหน้าต่างบริเวณปุ่ม Code ให้เราเลือกจะทำ Clone ด้วยวิธีใด ซึ่งทั้งหมด 3 วิธี คือ

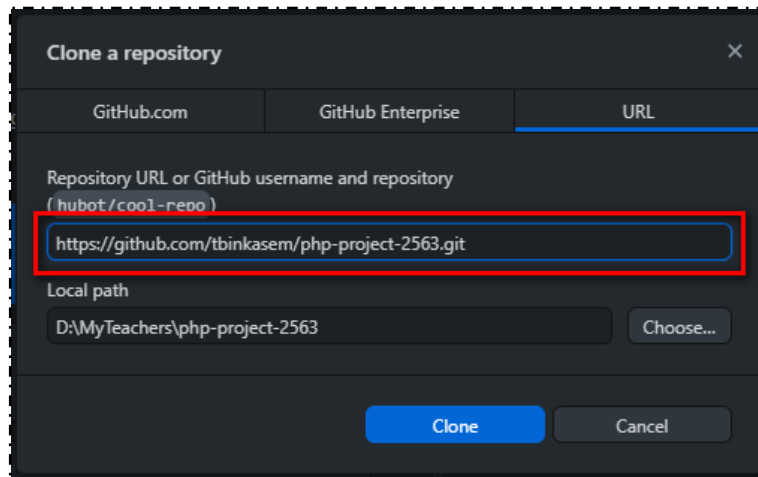
- a) HTTPS เป็นการ Clone โดยใช้โปรแกรมช่วย เช่น GitHub Desktop เป็นต้น
- b) SSH เป็นการ Clone โดยใช้ public SSH keys ซึ่งมีวิธีการที่ซับซ้อน
- c) GitHub CLI เป็นการ Clone โดยใช้คำสั่ง git clone URL ผ่านโปรแกรมจำพวก

Terminal (MAC) หรือ Command Prompt (Windows) เป็นต้น

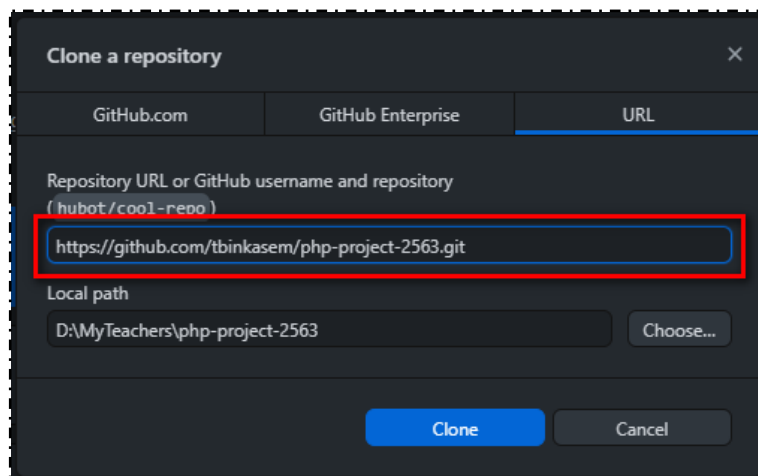
ในที่นี้เราใช้โปรแกรม GitHub Desktop ดังนั้นเราจะใช้ Clone แบบ HTTPS โดยการคลิกปุ่ม คัดลอก หรือ คลิกที่เมนู Open with GitHub Desktop ดังรูป



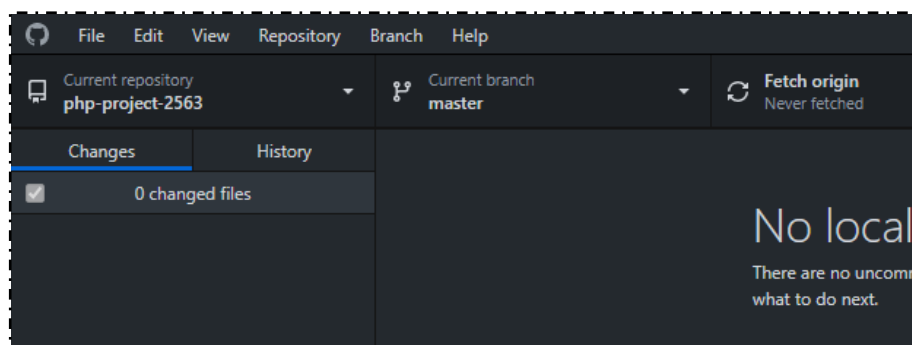
หากเราคลิกปุ่มคัดลอกตามกรอบสีแดง ให้เราไปวางลงในช่อง URL แล้วคลิกปุ่ม Clone ดังรูป



หากเราคลิกเมนู Open with GitHub Desktop คัดลอกตามกรอบสีแดง คอมพิวเตอร์ที่เราใช้งานจะทำการโปรแกรม GitHub Desktop ในเครื่องเราและใส่ URL ลงในช่องโดยอัตโนมัติ (หากเครื่องไม่ได้ลงโปรแกรม GitHub Desktop ไม่สามารถจะทำได้) ดังรูป



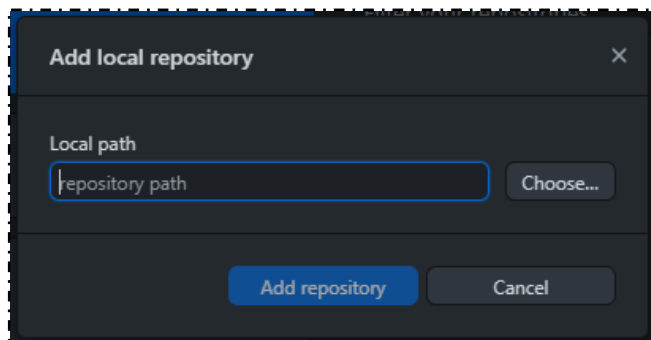
หลังจากนั้นเราก็กดปุ่ม Clone โปรแกรมก็จะสร้างจนเสร็จ จะปรากฏหน้าต่างโปรแกรม ดังรูป ซึ่งการทำงานหลังจากนี้ เหมือนเราใช้งาน Repository ที่เราสร้างขึ้นมาเองใหม่ โดยจะทำให้ Repository ของเราจะมีอยู่มากกว่า 1 ตำแหน่ง แต่จะเชื่อมโยงกัน หากมีการเปลี่ยนแปลงข้อมูลก็จะเปลี่ยนได้เหมือนเป็นเครื่องเดียวกัน ดังรูป



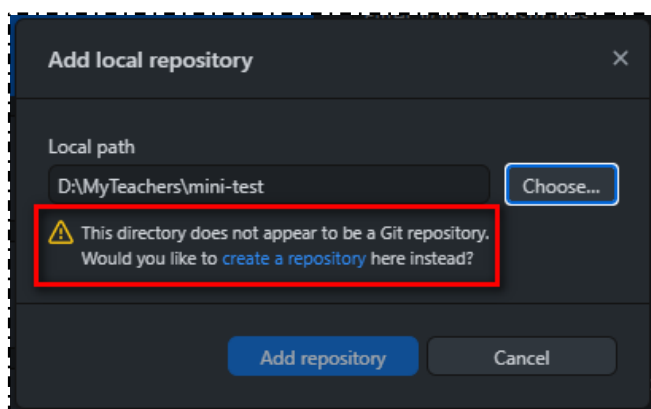
5.4 Add an Existing Repository from your hard drive เป็นการนำเอาโฟลเดอร์ที่ทำสร้างเป็น

Repository แล้ว แต่ไม่มีโปรแกรม GitHub Desktop หากเราต้องการเพิ่มมีขั้นตอน ดังนี้

5.3.1 คลิกที่ปุ่ม Add an Existing Repository from your hard drive จะปรากฏหน้าต่าง ดังรูป

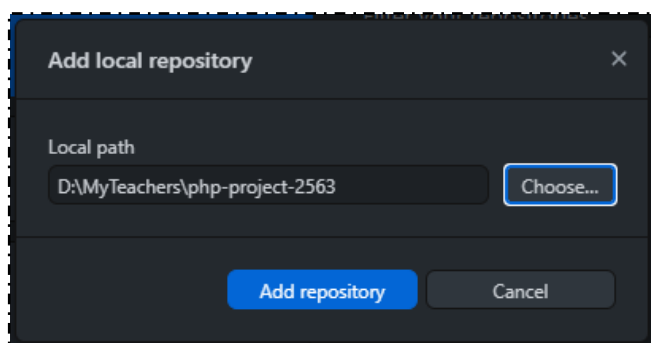


ให้เราคลิกปุ่ม Choose แล้วค้นหาโฟลเดอร์ที่เป็น Repository เดิมที่เคยทำไว้ ซึ่งหากโฟลเดอร์นั้นไม่ได้เป็นตำแหน่งของ Repository จะมีข้อความแจ้งเตือนให้ทราบ ดังรูป



ซึ่งเราสามารถจะสร้างโฟลเดอร์นั้นเป็น Repository ได้ทันที โดยการคลิกที่ข้อความสีฟ้าที่ชื่อ “create a repository” ก็จะเข้าสู่กระบวนการสร้าง Repository ใหม่ เหมือนในข้อ 5.2

แต่ถ้าโฟลเดอร์ที่เราเลือกเป็นตำแหน่งของ Repository หน้าต่างโปรแกรมจะปรากฏ ไม่มีข้อความแจ้งเตือนใด ๆ ดังรูป



จากนั้นให้ทำการกดปุ่ม Add repository จะเข้าสู่การใช้งาน Repository นั้นในโปรแกรม GitHub Desktop เหมือนในข้อ 5.2 และ 5.3