

ภาคผนวก ก
คู่มือการติดตั้งภายในระบบ

ระบบสารสนเทศเพื่อการบริหารจัดการเสาไฟฟ้าส่องสว่าง
ชุมชนเทศบาลตำบลหนองควาย อำเภอหางดง จังหวัดเชียงใหม่

คู่มือการติดตั้งโปรแกรมจำเป็นสำหรับใช้งานระบบสารสนเทศเพื่อการบริหารจัดการเสาไฟฟ้าส่อง

สว่าง เทศบาลชุมชนตำบลหนองควาย อำเภอหางดง จังหวัดเชียงใหม่

การติดตั้งโปรแกรมที่จำเป็น มีทั้งหมด 2 ส่วนคือ

- 1) คู่มือสำหรับการติดตั้งโปรแกรมจำลองเซิร์ฟเวอร์ (Docker)
- 2) การเปลี่ยนคีย์ Google API ก่อนใช้งาน
- 3) คู่มือการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อเข้าไปจัดการข้อมูลไฟล์ต่างๆบนเซิร์ฟเวอร์

คู่มือสำหรับการติดตั้งโปรแกรมจำลองเซิร์ฟเวอร์ (Docker)

1. การเริ่มต้นใช้งาน Docker เบื้องต้น

เริ่มจากเราต้องทำการติดตั้ง Docker โดยการติดตั้งเพียงแค่ว่าดาวน์โหลด Package มาติดตั้งเหมือนการติดตั้งโปรแกรมทั่วไปในคอมพิวเตอร์ ติดตั้งเพียงไม่กี่ขั้นตอนก็สามารถใช้งานได้เลย มีให้ติดตั้งทั้ง Mac, Windows, และ Linux มีวิธีติดตั้งตามลิงค์ด้านล่างนี้

- [Docker for Mac](#)
- [Docker for Windows](#)

ถ้าหากใช้ Windows รุ่น Home จะไม่สามารถติดตั้ง Docker for Windows ได้ (เนื่องจากต้องใช้ Microsoft Hyper-V ในการรัน)

ให้เปลี่ยนไปติดตั้ง [Docker Toolbox](#) แทน

- [Docker on Linux](#)

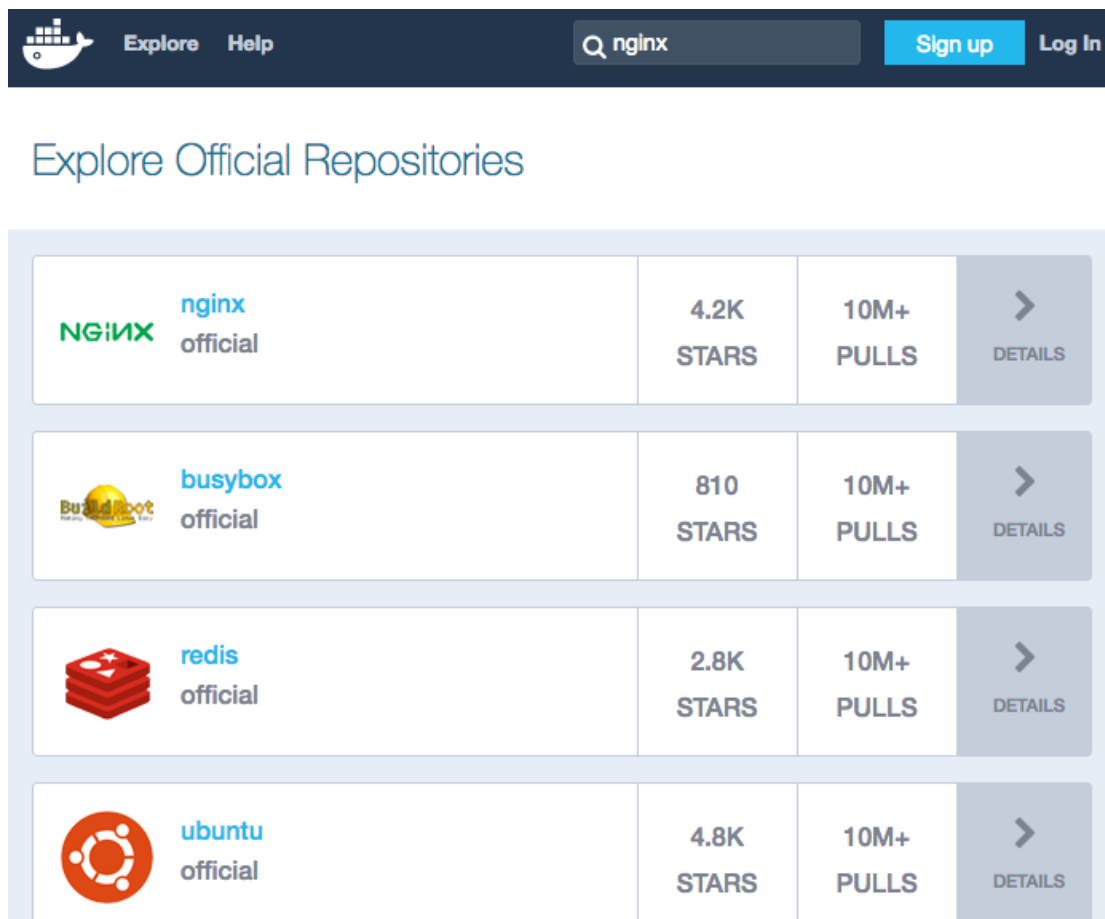
2. คำสั่ง Docker เบื้องต้น

สำหรับการใช้งาน Docker นั้นจะต้องใช้คำสั่งในการสั่งงาน ผ่าน Command Line เมื่อติดตั้งเสร็จแล้ว ให้เปิดโปรแกรม Terminal หรือ cmd แล้วมาเริ่มกันที่การใช้คำสั่ง Docker หลังจากติดตั้ง docker แล้ว ลองตรวจสอบเลขเวอร์ชัน ด้วยคำสั่ง

```
docker --version หรือ docker -v
```

3. เริ่มจากการ Pull Docker Image

ก่อนจะทำการสร้างและรัน Docker container ลองค้นหาใน Docker Hub ก่อนว่ามีอะไรให้เราใช้บ้าง ดังภาพที่ ก.1



ภาพที่ ก.1 Docker Hub ที่สามารถใช้งานได้

เมื่อเจอ Docker image ที่ต้องการ สามารถโหลด Docker image ที่ต้องการด้วยคำสั่งด้านล่างนี้

```
docker pull
```

ตัวอย่าง : ในที่นี่จะโหลด Docker image ที่ชื่อว่า nginx ให้ใช้คำสั่ง

```
docker pull nginx หรือ docker pull nginx:stable
```

หรือ

```
docker pull nginx:1.10
```

เราสามารถเลือกเวอร์ชันของ image ได้ด้วยการระบุ tag ลงไป หากไม่ระบุ tag ก็จะเป็นการเลือกใช้เวอร์ชันล่าสุด เช่น nginx ก็จะมี Docker Image ให้เลือกหลายๆเวอร์ชัน nginx:stable, nginx:1.10

4. การสร้าง Container

มาถึงขั้นตอนสำคัญ ก็คือการสร้าง Container โดยใช้คำสั่ง docker run เพื่อสร้าง Docker Container

```
docker run --name demo -d -p 8080:80 -v /home/me:/home/docker image
```

ตัวอย่าง

```
docker run --name myweb -p 8080:80 -v /myweb:/var/www/html -d nginx
```

คำสั่งจะค่อนข้างยาว เพราะคำสั่ง docker run จะต้องมีการใส่ option เข้าไปด้วย

--name demo คือ การตั้งชื่อให้กับ container โดยเมื่อเราใช้คำสั่ง start stop หรือ rm สามารถสั่งงาน container ได้จากชื่อ container ได้เช่นกัน **-d** เป็นการสั่ง container ให้รันแบบ background **-v /home/me:/home/docker** คือการ mount volume หรือเป็นการแชร์ไฟล์ระหว่าง container กับเครื่องเราให้สามารถเรียกใช้ไฟล์ร่วมกันได้ **-p 8080:80** เป็นการ map port ระหว่างเครื่อง และ container เช่น อย่ง container ของ nginx จะใช้ port 80 เป็น default หากต้องการให้เครื่องใช้ port **8080** เราก็กำหนดให้เป็น **-p 8080:80** **image** ชื่อของ Docker image ที่เราต้องการเรียกใช้ (จากตัวอย่างจะใช้ **nginx** เป็น image)

เมื่อมีการสั่ง docker run จะมีตรวจสอบว่ามี image นั้นอยู่ในเครื่องหรือไม่ ถ้าหากไม่พบ Docker ก็จะทำกร Pull Image นั้นๆจาก Docker Hub ลงมาให้การใส่ Option ต่างๆสำหรับแต่ละ Image จะไม่เหมือนกัน มีการ map port หรือ mount volume ต่างกัน โดยเราสามารถดูได้จากคำอธิบายของแต่ละ Image ได้ในเว็บ hub.docker.com

Port ที่เรากำหนดด้วย -p สามารถเข้าถึงได้จาก IP Address ของเครื่อง (localhost, 127.0.0.1 หรือ IP ในวงแลน) ตามด้วย Port ที่เรากำหนดไว้ ในที่นี้คือ <http://localhost:8080/> (ถ้าใช้ Docker Toolbox จะเป็น IP : 192.168.99.100 โดย default)

โดย Container ที่เราลองรันอยู่เป็น Web Server เพราะฉะนั้นถ้ามาลองเปิดดูใน browser จะพบว่าสามารถใช้งานได้แล้ว ดังภาพที่ ก.2



ภาพที่ ก.2 Web Server ที่พร้อมใช้งานแล้ว

5. ตรวจสอบสถานะ Container

เมื่อใช้คำสั่ง `docker run` เพื่อสร้างและรัน docker แล้ว โดยทำการลองตรวจสอบว่ามี Container ตัวไหนรันอยู่บ้าง ดังภาพที่ ก.3

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
388388f67777	nginx	"nginx -g 'daemon off'"	4 seconds ago	Up 2 seconds	443/tcp, 0.0.0.0:8080->80/tcp	myweb

ภาพที่ ก.3 ตัวอย่างการใช้คำสั่ง `docker ps`

จะสังเกตเห็นได้ว่ามี Container ที่ชื่อว่า `myweb` กำลังทำงานอยู่ โดยทุกๆ Container ก็จะมี Container ID เป็นของตัวเอง และเราสามารถสั่งงาน `start`, `stop`, `rm` ได้ด้วยการเรียกผ่าน Container ID ดังภาพที่ ก.4

```
CONTAINER ID    IMAGE
388388f67777   nginx
```

ภาพที่ ก.4 Container ID ของแต่ละตัว

หรือถ้าหากต้องการดูรายการ container ทั้งหมด ไม่ว่าจะรันอยู่และหยุดทำงานไปแล้วโดยใช้คำสั่ง

```
docker ps -a
```

6. หยุด Container

เมื่อต้องการให้ container หยุดทำงานจะต้องใช้คำสั่ง docker stop ตามด้วย Container ID หรือ Name ที่ใช้อ้างอิง Container

```
docker stop a715
```

ในที่นี้จะใช้ Container ID ในการอ้างอิงเพื่อสั่งหยุด container ที่ต้องการ (การเรียกใช้ Container ID ไม่จำเป็นต้องใส่ให้ครบทุกตัวอักษรก็ได้)

7. ลบ Container

เมื่อไม่ใช้งาน Container นั้นแล้ว ไม่อยากจะใช้ต่อสามารถลบ Container ได้ด้วยคำสั่ง

```
docker rm a715
```

จะใช้คำสั่งนี้ได้ก็ต่อเมื่อ Container นั้นหยุดทำงานไปแล้ว

8. ดู Docker Image ทั้งหมด ดังภาพที่ ก.5

```
docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	ba6bed934df2	6 days ago	181.4 MB
centos	6	f07f6ca555a5	3 weeks ago	194.6 MB
dockerlemp_phpfp	latest	65aec54003c8	3 weeks ago	293.9 MB
dockertest_phpfp	latest	9899cc5a98ab	3 weeks ago	293.9 MB
mariadb	10.1	7e149af02fc0	4 weeks ago	391.9 MB
httpd	latest	fe3336dd034d	4 weeks ago	195.4 MB
httpd	2.2	563e7de6300a	4 weeks ago	185.8 MB
php	7.0-fpm-alpine	cd228f10875c	5 weeks ago	70.6 MB
nginx	stable-alpine	c07e2bb859e2	5 weeks ago	54.68 MB

ภาพที่ ก.5 การแสดงผลหลังจากใช้งานคำสั่ง docker images

9. คำสั่งลบ Image

เมื่อต้องการลบ image จะต้องแน่ใจก่อนว่ามี container ไหนเรียกใช้ image นี้อยู่รึเปล่านั้น ถ้าไม่มีการเรียกใช้งานก็สามารถสั่งลบ image ด้วยคำสั่ง docker rmi ตามด้วย Image หรือ Image name ที่อ้างอิงกับ Image นั้นๆ

```
docker rmi c07e หรือ docker rmi nginx:stable-alpine
```

10. คำสั่งการสร้างไฟล์ docker-compose.yml ไว้ในโปรเจกต์โฟลเดอร์ และใส่โค้ดดังนี้

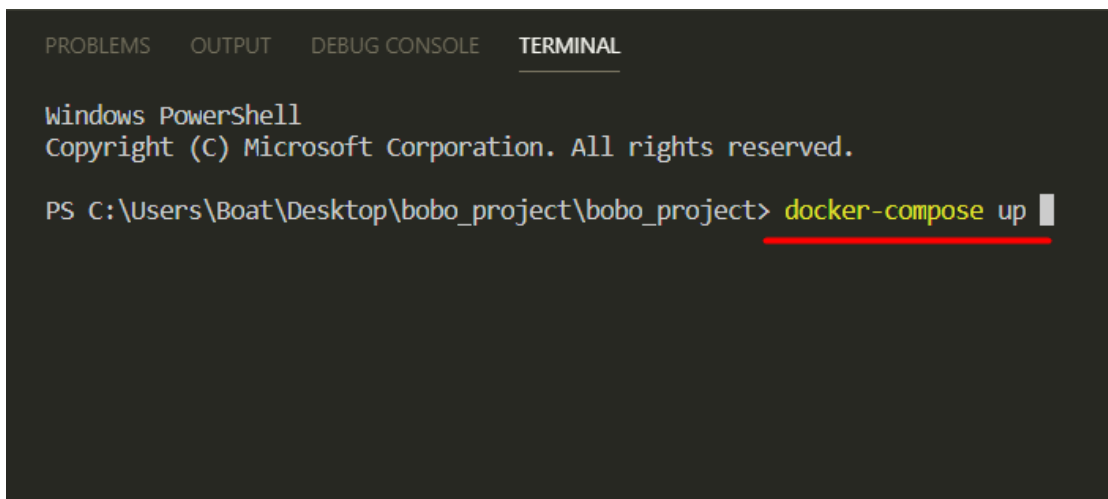
```

version: '3'
services:
web:
build: .
ports:
- "3000:3000"
volumes:
- ./app:/nodeapp
redis:
image: "redis:3.0"

```

11. สั่ง start จากโปรเจกต์โฟลเดอร์ หรือสั่งผ่าน Terminal ของโปรแกรมได้เลย โดยใช้คำสั่งดังภาพที่ ก. 6

```
$ docker-compose up
```



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

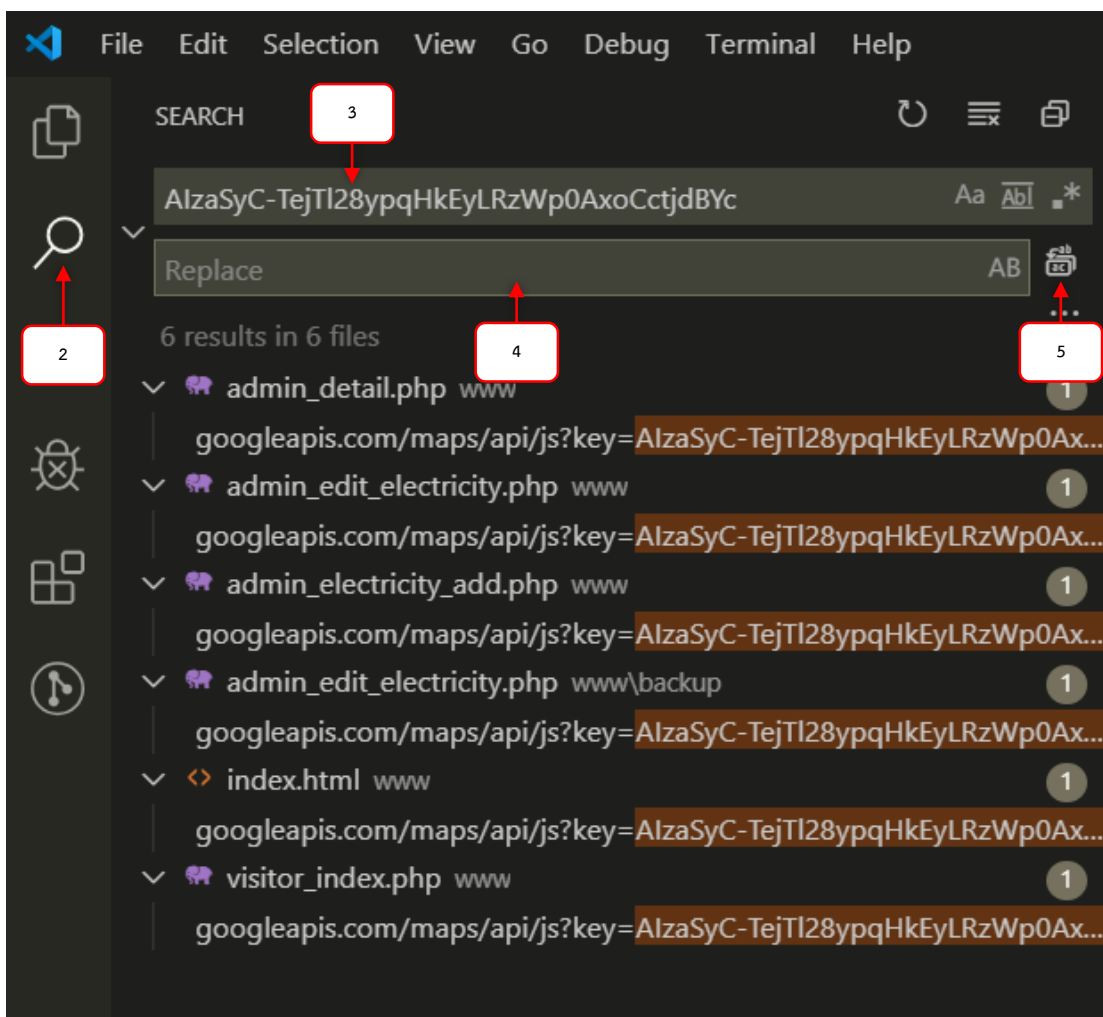
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Boat\Desktop\bobo_project\bobo_project> docker-compose up

```

ภาพที่ ก.6 การ start ผ่าน Terminal จากโปรแกรม visual studio code

คู่มือสำหรับการเปลี่ยนคีย์ Google API ก่อนใช้งาน

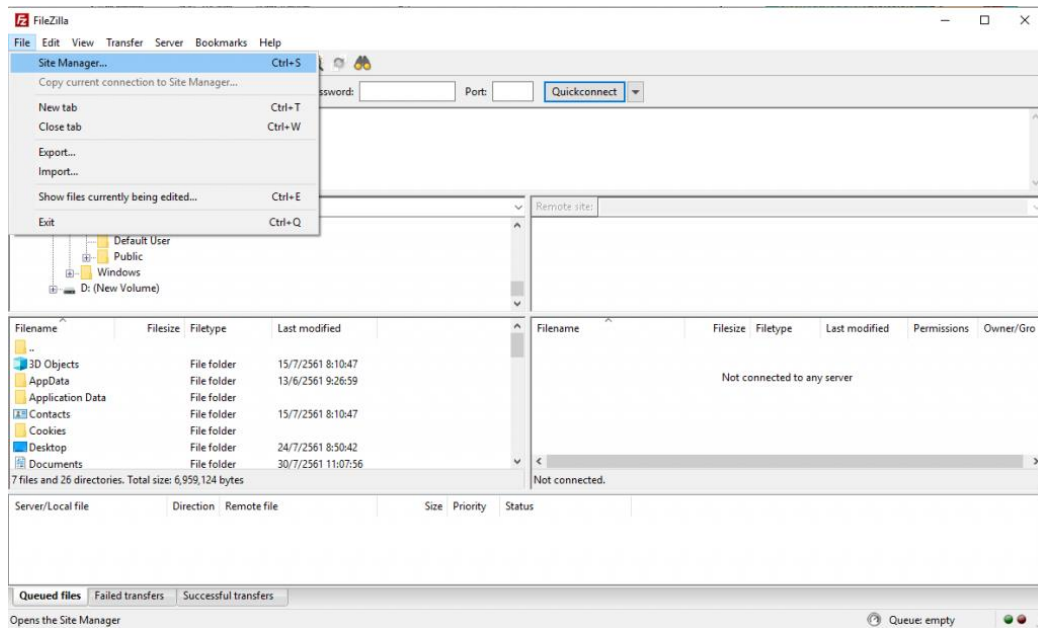


ภาพที่ ก.7 คู่มือสำหรับการเปลี่ยนคีย์ Google API ก่อนใช้งาน

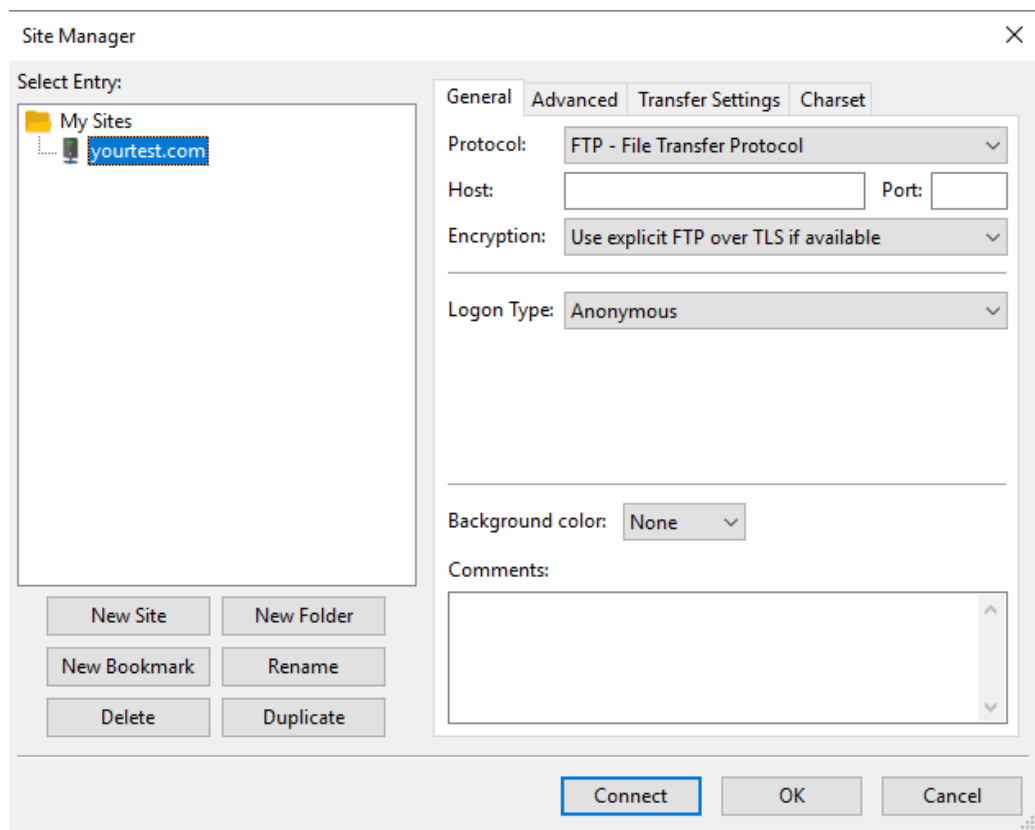
1. เริ่มต้นโดยการเปิด Source code โปรแกรมขึ้นมา โดยในตัวอย่าง ผู้เขียนได้ใช้โปรแกรม Visual Studio Code
2. ทำการไปที่เมนู Search หรือคีย์ลัด (Ctrl+Shift+F)
3. ในช่อง Search ทำการกรอก Code ของ Google Key API ชุดเก่าเข้าไป ตามตัวอย่าง

AlzaSyC-TejTl28ypqHkEyLRzWp0AxoCctjdBYc

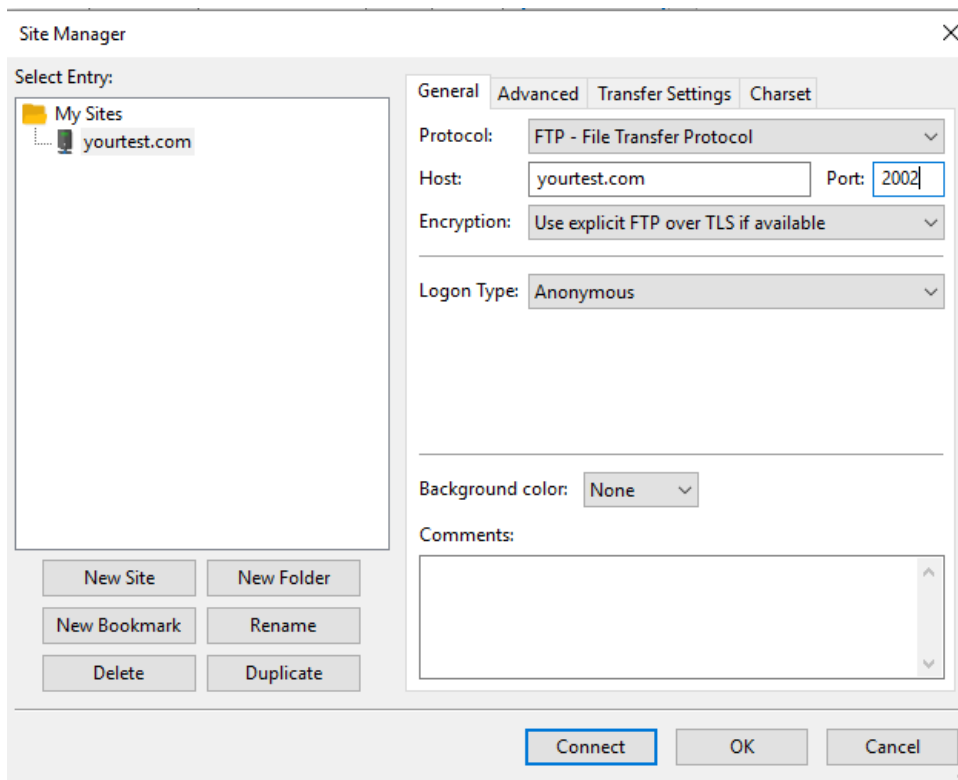
4. และทำการแทนที่ Google Key API ชุดใหม่เข้าไปในช่อง Replace
 5. จากนั้นกดปุ่ม Replace All หรือคีย์ลัด (Ctrl+Alt+Enter) เพื่อเสร็จสิ้นขั้นตอน
- คู่มือการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อเข้าไปจัดการข้อมูลไฟล์ต่างๆบนเซิร์ฟเวอร์



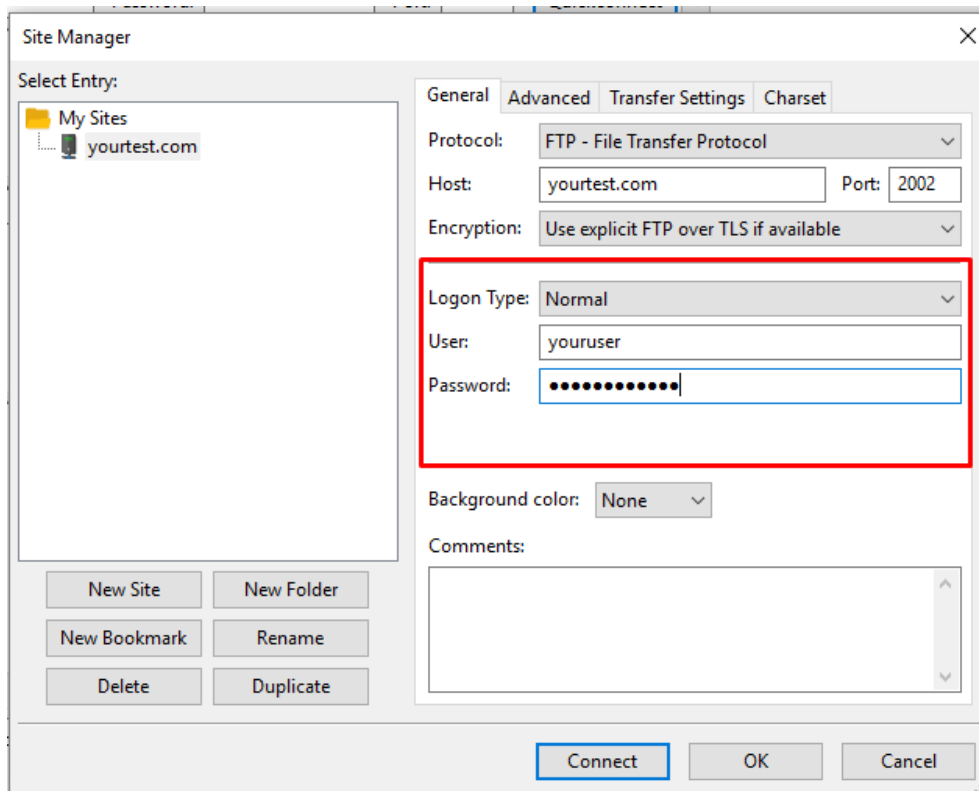
ภาพที่ ก.8 เปิดเมนู File > Site Manager จากนั้นคลิกที่ New Site



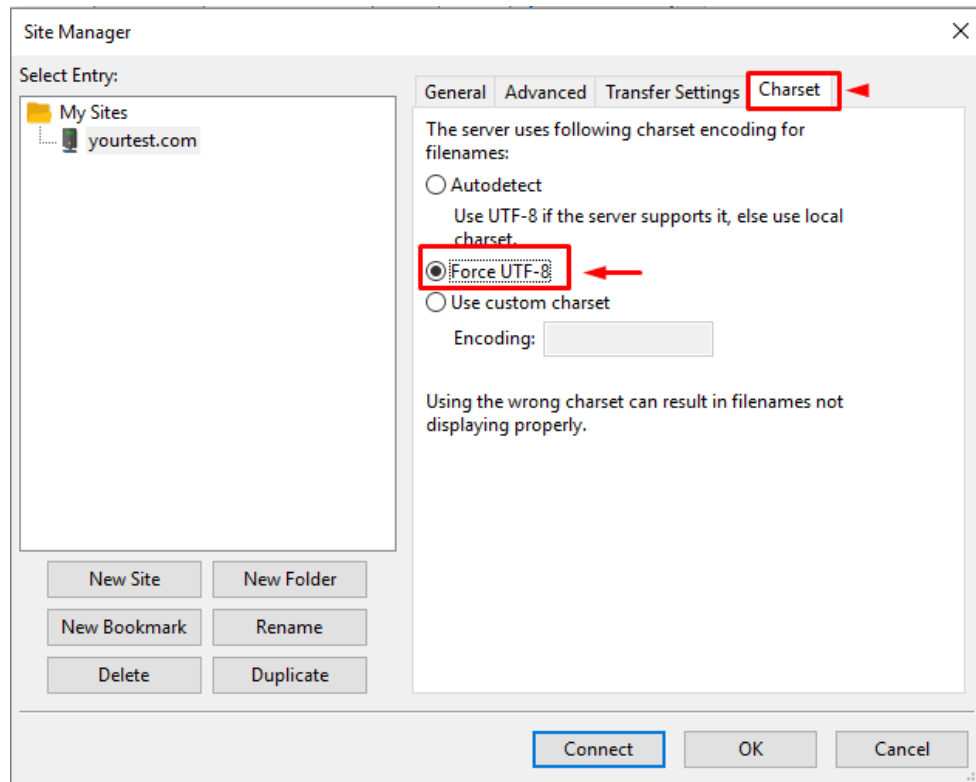
ภาพที่ ก.9 กำหนดชื่อเว็บไซต์



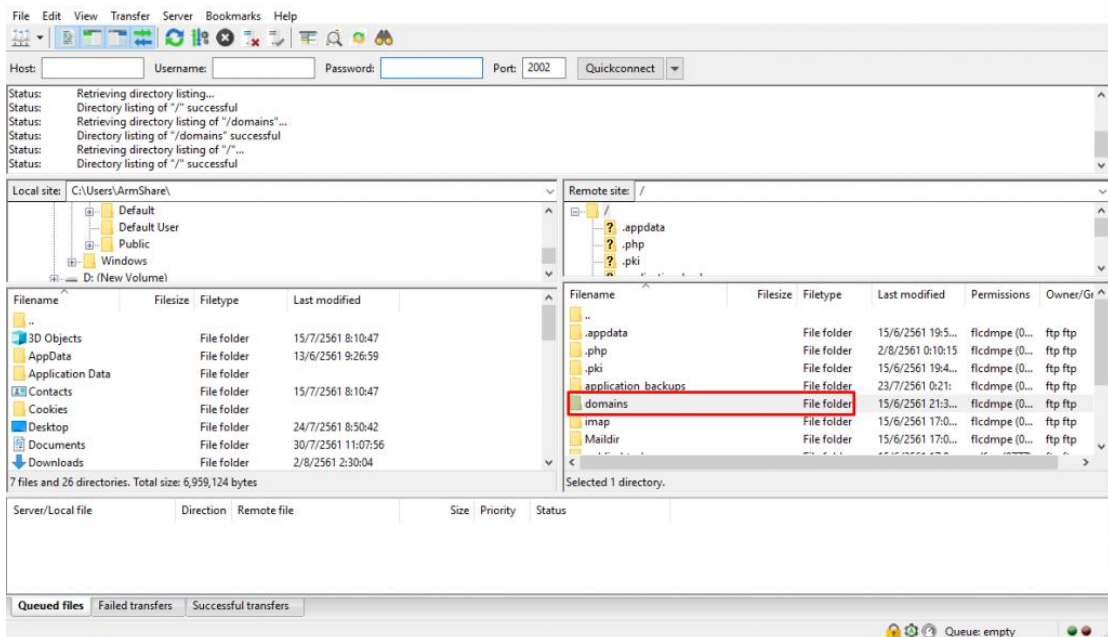
ภาพที่ ก.10 ระบุรายละเอียดชื่อโดเมนในช่อง Host



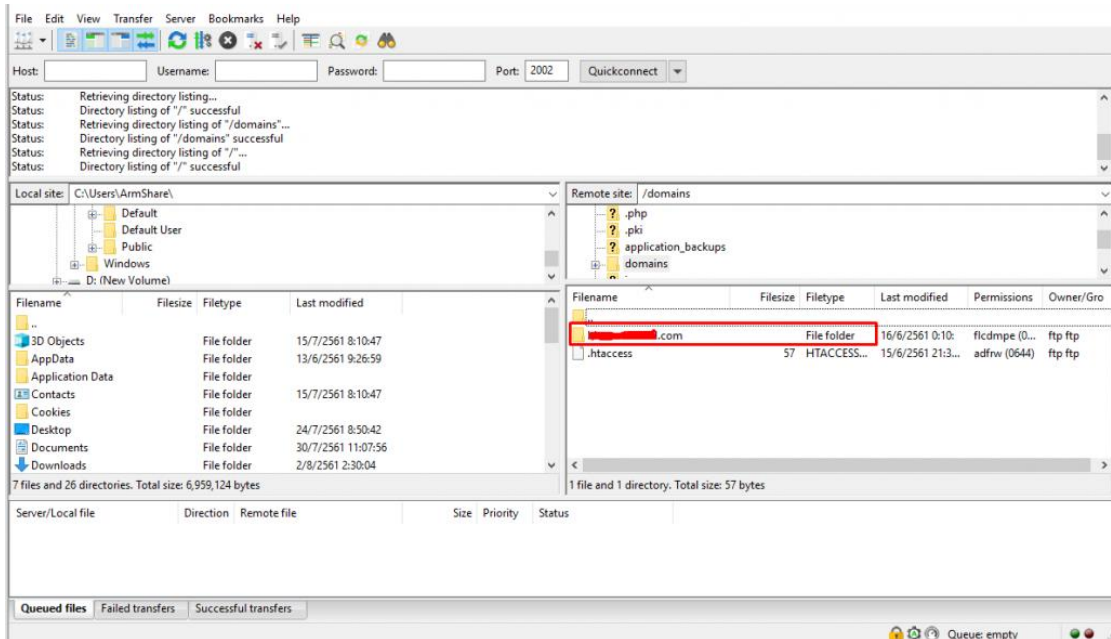
ภาพที่ ก.11 เลือก Logon Type เป็น Normal และกรอก FTP: Username และ Password



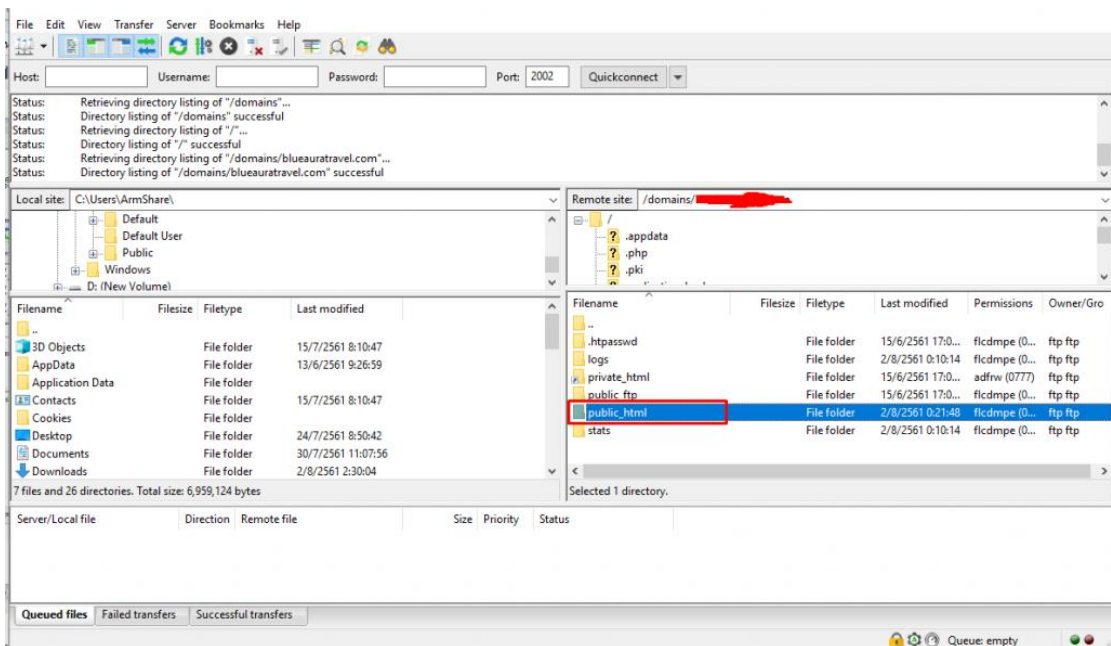
ภาพที่ ก.12 คลิกที่ Tab Charset เลือก Force UTF-8 และกดปุ่ม connect



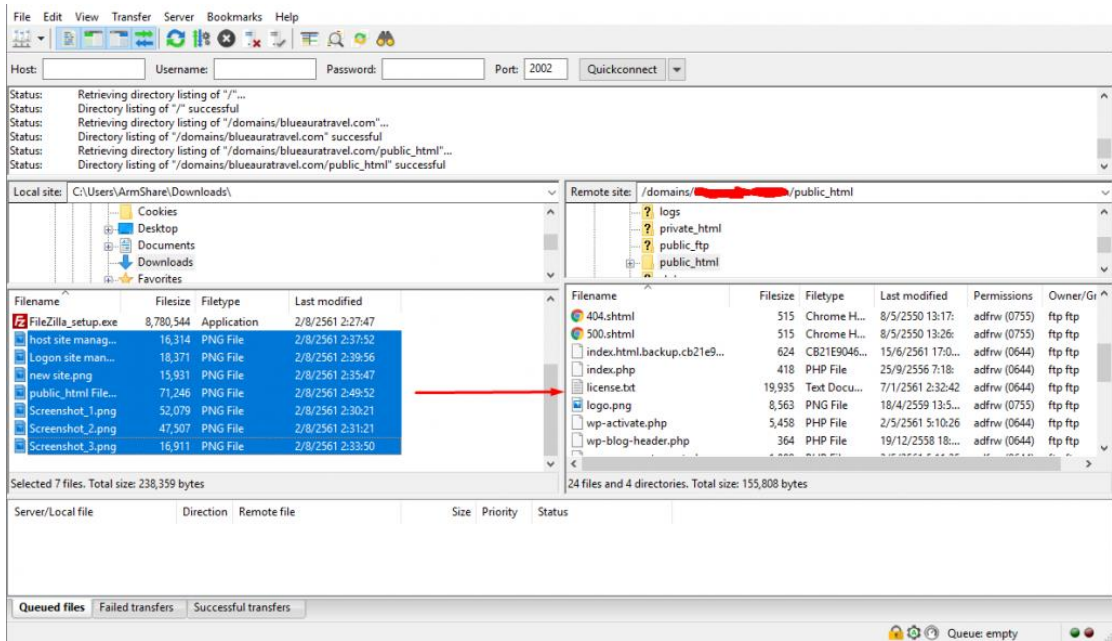
ภาพที่ ก.13 ดับเบิลคลิกที่ Directory domains ด้านขวามือ



ภาพที่ ก.14 คลิกเข้าไปที่ชื่อโดเมนของคุณ



ภาพที่ ก.15 ดับเบิลคลิกที่ public_html



ภาพที่ ก.16 ทำการอัปโหลดด้วยการลากไปวางทางฝั่งขวา