File Server & Task Management Functions Integrated with Microsoft Azure

(โครงงานสหกิจศึกษา)

นายศิรินันท์ การบรรจง

โครงงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรเทคโนโลยีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2558

ชื่อ : นายศิรินันท์ การบรรจง

ชื่อโครงงาน : File Server & Task Management Functions   
 Integrated with Microsoft Azure

สาขาวิชา : เทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร.ขนิษฐา นามี

ปีการศึกษา : 2560

**บทคัดย่อ**

โครงงานสหกิจนี้มีวัตถุประสงค์เพื่อศึกษาเรียนรู้ และเข้าใจเกี่ยวกับการทำ File Server Function และ Task Management Function โดยใช้แพลตฟอร์ม Microsoft Azure เป็นตัวช่วยในการดำเนินงาน เพื่อการจัดเก็บ เรียกใช้งานไฟล์ข้อมูลที่ได้มีการเก็บเอาไว้ใน File Server ที่ได้สร้างขึ้น ออกมาแสดงผลผ่านทาง Web Browser และยังมีในเรื่องของความปลอดภัยในการเข้าถึง ไฟล์ข้อมูลเนื่องจากมีการใช้ Azure Active Directory เพื่อจำกัดสิทธิของผู้ใช้งานรวมไปถึงยังมีการใช้ Azure content Delivery Network ที่ช่วยในการกระจายไฟล์ข้อมูลภายใน File Server ออกไปอยู่บน Azure Cloud Server เพื่อช่วยในการเข้าถึงไฟล์ข้อมูลได้อย่างรวดเร็วของผู้ใช้งานที่อยู่ในพื้นที่ห่างไกลหรือต่างประเทศ สำหรับการศึกษาและการทำ Task Management Function นั้นเป็นตัวช่วยในการแจ้งเตือนงานที่ต้องดำเนินการไปยัง Application มีทั้งหมด 3 รายการได้แก่ Trello, Microsoft To-Do, Microsoft Planner โดยมีวิธีการดำเนินงานคือ จะรับข้อมูลจากผุ้ใช้งานที่ส่งเข้ามาและนำไปประมวลผลในฟังก์ชันเพื่อไปสร้างรายการ Task บน Application ที่ได้กล่าวไว้ข้างต้น   
 ในการทำงานของ File Server Function นั้นจะช่วยแก้ปัญหาด้านการทดลองเว็บไซต์ขนาดเล็กที่ปกติต้องทำโดยวิธีที่เปลืองทรัพยากรมากกว่านี้ และในส่วนของ Task Management function นั้นจะเป็นการช่วยในการแจกจ่ายงาน, แบ่งหน้าที่ของงานให้แก่พนักงานภายในบริษัท

คำสำคัญ : Azure, File Server, Function

ผศ.ดร.ขนิษฐา นามี อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

**กิตติกรรมประกาศ**

โครงงานสหกิจศึกษาเรื่อง “File Server & Task Management Functions Integrated with Microsoft Azure” สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับคำแนะนำจากท่านอาจารย์ที่ปรึกษาโครงงานสหกิจ ผศ.ดร.ขนิษฐา นามีและขอบคุณคณาจารย์ประจำสาขาวิชาเทคโนโลยีสารสนเทศทุกท่าน ที่คอยติดตามและให้คำแนะนำในการทำโครงงานสหกิจศึกษาในครั้งนี้ ขอขอบคุณทางบริษัท Together Teamsolutions Co., Ltd. ที่ได้ให้คำแนะนำ และให้ความรู้ถึงข้อมูลที่ใช้ในการทำโครงงานสหกิจศึกษาในครั้งนี้ สุดท้ายนี้ขอราลึกถึงพระคุณของบิดามารดา ครูบาอาจารย์ของข้าพเจ้าที่ได้อบรมสั่งสอนจนถือได้ว่าประสบความสำเร็จไปอีกก้าวหนึ่งของชีวิต ขอขอบคุณน้ำใจจากบุคลากรทุกท่านในภาควิชาเทคโนโลยีสารสนเทศที่ให้คำแนะนำรวมไปถึงเหล่าเพื่อน และ พี่ ทุกท่านที่ช่วยเหลือและเป็นส่วนหนึ่งที่ทำให้เกิดความสำเร็จของการจัดทำโครงงานสหกิจศึกษาในครั้งนี้

ศิรินันท์ การบรรจง

**สารบัญ**

[บทที่ 1 **บทนำ** 6](#_Toc520752986)

[1. ข้อมูลพื้นฐานการปฏิบัติงานสหกิจศึกษา 8](#_Toc520752987)

[2. ความเป็นมาและปัญหา 8](#_Toc520752988)

[3. วัตถุประสงค์ของโครงงาน 10](#_Toc520752989)

[4. ขอบเขตโครงงาน 10](#_Toc520752990)

[5. ประโยชน์ที่คาดว่าจะได้รับ 14](#_Toc520752991)

[บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 15](#_Toc520752992)

[Microsoft Azure Function App 15](#_Toc520752993)

[Azure logic app 15](#_Toc520752994)

[Azure Active Directory 15](#_Toc520752995)

[Azure Content Delivery Network 17](#_Toc520752996)

[Visual Studio Code 17](#_Toc520752997)

[C# 18](#_Toc520752998)

[JavaScript 20](#_Toc520752999)

[Json 21](#_Toc520753000)

[API 22](#_Toc520753001)

[บทที่ 3 ขั้นตอนการพัฒนาและการออกแบบ 23](#_Toc520753002)

[**1.** ศึกษาการใช้งาน Microsoft Azure Function App 34](#_Toc520753003)

[**2.** ศึกษาการใช้งาน Microsoft Azure Function App 39](#_Toc520753004)

[**3.** ศึกษาการทำหน้าฟอร์ม โดยใช้ Form.IO 41](#_Toc520753005)

[**4.** ศึกษาวิธีการนำฟอร์มที่ได้ทำการรันในไฟล์ HTML นำไปเก็บไว้ใน File Server 42](#_Toc520753006)

[**5.** ศึกษาวิธีการทำ Azure logic app 44](#_Toc520753007)

[บทที่ 4 **ผลการดำเนินงาน** 45](#_Toc520753008)

[1. สามารถเรียกใช้งานไฟล์ผ่าน File Server Function ได้ 45](#_Toc520753009)

[2. สามารถเรียกใช้งานไฟล์ผ่าน File Server Function โดยเรียกผ่าน CDN ได้ 46](#_Toc520753010)

[3. สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จ 48](#_Toc520753011)

[**4.** สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จเมื่อกรอกค่า card name ค่า 49](#_Toc520753012)

[**5.** สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จเมื่อกรอกข้อมูลครับยกเว้นusername 50](#_Toc520753013)

[บทที่ 5 สรุปผลการดำเนินงาน 53](#_Toc520753014)

[1. สรุปผลการดำเนินงาน 53](#_Toc520753015)

[2. ปัญหาและอุปสรรค 53](#_Toc520753016)

[3. ข้อเสนอแนะ 53](#_Toc520753017)

**สารบัญรูปภาพ**

[ภาพที่ 1ตัวอย่างการกระบวนการของ Function App 15](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755597)

[ภาพที่ 2 ตัวอย่างการกระบวนการของ Logic App 15](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755598)

[ภาพที่ 3 ตัวอย่างการกระบวนการของ Microsoft Azure Active Directory 16](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755599)

[ภาพที่ 4 ตัวอย่างการกระบวนการของ Microsoft Content Delivery Network 17](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755600)

[ภาพที่ 5 ภาพตัวอย่างของโปรแกรม Visual Studio Code 18](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755601)

[ภาพที่ 6 ภาพสัญลักษณ์ของโปรแกรม Visual Studio Code 18](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755602)

[ภาพที่ 7ภาพสัญลักษณ์ของ C# 18](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755603)

[ภาพที่ 8 ภาพตัวอย่างของ JavaScript 21](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755604)

[ภาพที่ 9 ภาพสัญลักษณ์ของ JavaScript 21](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755605)

[ภาพที่ 10 ภาพตัวอย่างข้อมูล Json 22](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755606)

[ภาพที่ 11 ภาพตัวอย่างของ API 22](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755607)

[ภาพที่ 12 แสดงตัวอย่างคำสั่งในหน้า File Server Function 34](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755608)

[ภาพที่ 13 แสดงตัวอย่างในหน้า Proxies 35](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755609)

[ภาพที่ 14 แสดงตัวอย่างการเปิดใช้งาน Authentication / Authorization 36](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755610)

[ภาพที่ 15 แสดงตัวอย่างหน้าไฟล์ Authorization 36](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755611)

[ภาพที่ 16 แสดงตัวอย่างหน้า Login เข้าใช้งานไฟล์ 37](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755612)

[ภาพที่ 17 แสดงตัวอย่างหน้าที่ใช้สร้าง CDN 37](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755613)

[ภาพที่ 18 แสดงตัวอย่างหน้า Endpoint 38](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755614)

[ภาพที่ 19 แสดงตัวอย่างหน้าการใช้งานจากลิงค์(จะเห็นว่าลิงค์นั้นเป็น .azureedge.net) 38](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755615)

[ภาพที่ 20 แสดงตัวอย่างหน้าฟอร์มในการรับข้อมูล 39](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755616)

[ภาพที่ 21 แสดงตัวอย่างของ Trello 40](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755617)

[ภาพที่ 22 แสดงตัวอย่างของ components ใน Form.IO 41](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755618)

[ภาพที่ 23 แสดงตัวอย่างของคำสั่งปุ่ม submit ใน Form.io 41](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755619)

[ภาพที่ 24 แสดงตัวอย่างการเรียกหน้าฟอร์มผ่านไฟล์ HTML 42](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755620)

[ภาพที่ 25 แสดงตัวอย่างไฟล์ที่อัพโหลดลง File Server Function 42](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755621)

[ภาพที่ 26 แสดงตัวอย่างการเรียกไฟล์เพลง 43](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755622)

[ภาพที่ 27 แสดงตัวอย่าง Logic App เชื่อมต่อกับ File Server Function 44](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755623)

[ภาพที่ 28 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง 45](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755624)

[ภาพที่ 29 แสดงตัวอย่างการเรียกใช้งานไฟล์รูปภาพ 46](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755625)

[ภาพที่ 30 แสดงตัวอย่างการเรียกใช้งานไฟล์ index.html 46](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755626)

[ภาพที่ 31 แสดงตัวอย่างการเรียกใช้งานไฟล์ index.html ผ่าน CDN 47](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755627)

[ภาพที่ 32 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง ผ่าน CDN 47](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755628)

[ภาพที่ 33 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง ผ่าน CDN 48](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755629)

[ภาพที่ 34 แสดงตัวอย่างการกรอก input แล้วสำเร็จ 48](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755630)

[ภาพที่ 35 แสดงตัวอย่าง card ใน Trello ตามข้อมูลที่ได้รับ 49](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755631)

[ภาพที่ 36 แสดงตัวอย่างการกรอก input เฉพาะ card name 49](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755632)

[ภาพที่ 37 แสดงตัวอย่าง card ใน Trello ตามข้อมูลที่ได้รับ 50](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755633)

[ภาพที่ 38 แสดงตัวอย่าง Alert ของการกรอก input username ผิด 51](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755634)

[ภาพที่ 39 แสดงตัวอย่าง Alert ของการกรอก input username ผิด 51](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755635)

[ภาพที่ 40 แสดงตัวอย่าง Alert ของการกรอก input username ผิด 52](file:///C:\Users\MOSHI\Desktop\กฟดฟหกดฟหดฟหดฟหดด.docx#_Toc520755636)

# **บทที่ 1 บทนำ**

ข้อมูลพื้นฐานการปฏิบัติงานสหกิจศึกษา

* 1. ชื่อสถานประกอบการ บริษัท Together Teamsolutions Co., Ltd.

ที่อยู่ 330/30 ถนนพระตำหนัก ซอย 5 เมืองพัทยา อำเภอบางละมุง จังหวัดชลบุรี 20260

* 1. ลักษณะงานที่นักศึกษาได้รับมอบหมายให้รับผิดชอบ
* ทำงานที่ได้รับตามที่หัวหน้าทีมมอบหมาย
* เป็นการสร้างฟังก์ชัน,ทดสอบหน้าเว็บไซต์
* งานที่ทำนั้นจะเป็นในส่วนของ backend
* งานที่ทำเกือบทั้งหมดนั้นจะผ่านแพลตฟอร์ม Microsoft Azure
  1. พนักงานที่ปรึกษา และตำแหน่งงานของพนักงานที่ปรึกษา
* นายพอเจตร์ กุลจิตติญาณี ตำแหน่ง Team leader
  1. ระยะเวลาที่ปฏิบัติงาน
* วันที่ 4 มิถุนายน 2560 ถึง 30 พฤศจิกายน 2561

ความเป็นมาและปัญหา

เนื่องจากผมได้มีโอกาสฝึกงานกับทางบริษัท Together Teamsolutions Co., Ltd. และได้รับมอบหมายให้ศึกษาและใช้งาน “Microsoft Azure” แพลตฟอร์ม ซึ่งเป็นคลาวด์แพลตฟอร์มของทางบริษัท Microsoft จากการศึกษาพบว่า Microsoft Azure นั้นมีเซอร์วิสต่าง ๆ ที่ช่วยอำนวยความสะดวกแก่การพัฒนาแอพพลิเคชั่นมากมาย โดยสามารถสมัครใช้งานเพื่อทำการทดลองใช้ Microsoft Azure ได้ฟรีในเบื้องต้นภายในระยะเวลา 1 เดือน พร้อมกับเครดิตสำหรับใช้จ่ายเพื่อใช้บริการเซอร์วิสต่าง ๆในจำนวน $200

นอกจากนี้หากเป็นนักศึกษาแล้ว ทางบริษัท Microsoft ก็ได้มีส่วนของแอคเคาท์สำหรับนักศึกษาโดยเฉพาะสำหรับมหาวิทยาลัยที่ร่วมโครงการ DreamSpark ซึ่งมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ก็เป็นหนึ่งในมหาวิทยาลัยที่ร่วมโครงการนี้ โดยแอคเคาท์สำหรับนักศึกษา สามารถสมัครใช้งานได้ฟรีโดยไม่ต้องกรอกข้อมูลรายละเอียดบัตรเครดิต(บุคคลทั่วไปหากต้องการสมัครใช้งาน ต้องกรอกข้อมูลรายละเอียดบัตรเครดิตด้วย) และสามารถใช้งานเซอร์วิสหลายเซอร์วิสได้ฟรี โดยไม่เสียค่าใช้จ่ายตามที่บริษัท Microsoft ได้กำหนด ภายในระยะเวลา 1 ปี และยังได้รับเครดิตสำหรับใช้จ่ายเพื่อใช้บริการเซอร์วิสต่าง ๆ อีกจำนวน $100 ซึ่งถือว่าเพียงพอแล้วสำหรับการศึกษา เรียนรู้ Microsoft Azure แพลตฟอร์ม

ทางบริษัทมุ่งหวังให้ศึกษา Microsoft Azure เพื่อพัฒนา Function สำหรับใช้งานภายในองค์กรซึ่งประกอบไปด้วย Function ดังต่อไปนี้

1.File Server Function

2.Create Task in Trello

3.Create Task in Microsoft To-Do

4.Create Task in Microsoft Planner

File Server Function – โดยปกติแล้วเมื่อเราต้องการพัฒนาเว็บแอพพลิเคชั่นบน Microsoft Azure เราจะใช้งานตัวเซอร์วิสที่ชื่อว่า Web App เซอร์วิส ซึ่งเป็นตัวที่สร้างและแสดงผลเว็บแอพพลิเคชั่นโดยตรง แต่ในบางครั้งเราต้องการเพียงแค่แสดงผลแอพพลิเคชั่นที่มีขนาดไม่ใหญ่มากนัก การจะใช้งาน Web App เซอร์วิสโดยตรงนั้นจึงมองว่าเป็นการสิ้นเปลืองทั้งทรัพยากรพื้นที่ และ ค่าใช้บริการที่ต้องเสียไป จึงได้ทำการค้นหาวิธีที่จะสามารถมาทดแทนการทำงานของ Web App เซอร์วิสได้ และพบว่าโครงสร้างของ Function App ซึ่งใช้สำหรับพัฒนาฟังก์ชันต่าง ๆ บน บน Microsoft Azure นั้น มีโครงสร้างคล้ายคลึงกับตัว Web App เซอร์วิส เช่นกัน จึงเป็นที่มาของการสร้าง Function สำหรับเรียกใช้งานและแสดงผลไฟล์ที่ถูกเก็บไว้บน Function App ขึ้นมาทดแทน

File Server Function นี้จะถูกพัฒนาขึ้นมาทดแทนทั้งในส่วนของการจัดเก็บไฟล์ต่าง ๆ บนเซิร์ฟเวอร์ และส่วนของการแสดงผลไฟล์ที่ถูกจัดเก็บไว้ เพื่อทดแทนการใช้งาน Storage เซอร์วิสและ Web App เซอร์วิสบน Microsoft Azure

ในส่วนของ Create Task in Trello Function, Create Task in Microsoft To-Do และ Create Task in Microsoft Planner นั้น ถูกเรียกรวมว่า Task Management Function จะถูกพัฒนาเพื่อนำมาใช้งานภายในองค์กรในส่วนของการจัดการบริหารงานต่าง ๆ สำหรับพนักงานภายในบริษัท

การทำงานในบริษัทหรือองค์กร ต้องมีการทำงานเป็นทีม มีหัวหน้าเป็นคนดูแลคอยแจกจ่ายงาน และแบ่งงานให้ทำ โดยปกติแล้วจะต้องมีการประชุมแบ่งงานและมีการจดบันทึกในสิ่งที่ได้รับมอบหมาย ซึ่งก็อาจจะมีความผิดพลาดเกิดขึ้นได้ จากการสื่อสารที่ผิดพลาดและความเข้าใจที่ไม่ตรงกัน จึงได้มีความพยายามที่จะนำเอาระบบ Task Management เข้ามาใช้งานเพื่อช่วยจัดการในเรื่องนี้ โดยจะใช้งานเป็น Function สำหรับสร้างงาน มอบหมายงานและกระจายงานให้กับพนักงานที่เกี่ยวข้อง และสามารถติดตามสถานะงาน พูดคุย สอบถามปัญหาต่าง ๆ ที่เกิดขึ้นได้ ถือเป็นการลดระยะเวลาในการประชุม และติดตามสถานะการทำงานได้เป็นอย่างดี

วัตถุประสงค์ของโครงงาน

* + เพื่อศึกษา Microsoft Azure แพลตฟอร์ม, Function App Service, Logic Apps Service รวมถึงเซอร์วิสอื่น ๆ ที่มีอยู่บน Microsoft Azure และนำมาพัฒนาแอพพลิเคชั่นสำหรับการใช้งานภายในองค์กร
  + เพื่อพัฒนา File Server Function ให้สามารถเรียกหรือแสดงผลไฟล์ที่ถูกเก็บเอาไว้ภายในออกมาใช้งานได้
  + เพื่อพัฒนาระบบ Task Management สำหรับจัดการบริหารงานต่าง ๆ ภายในบริษัท
  + เพื่อศึกษาวิธีการกำหนดสิทธิ์ การเข้าถึงแอพพลิเคชั่น การเรียกใช้งานเซอร์วิสต่าง ๆ โดยใช้การกำหนดสิทธิ์ผ่าน Azure Active Directory
  + เพื่อศึกษาวิธีการป้องกันการเข้าถึง Function จาก Function App เซอร์วิสด้วยวิธี URL Authorization Rules
  + เพื่อศึกษาการใช้งาน Azure Content Delivery Network สำหรับการทำ Caching ข้อมูลจาก File Server Function และกระจายข้อมูลไปยัง Data Center อื่น ๆ เมื่อผู้ใช้งานเรียกใช้
  + เพื่อศึกษาการใช้งาน Form.IO สำหรับการออกแบบฟอร์มสำหรับระบบ Task Management

ขอบเขตโครงงาน

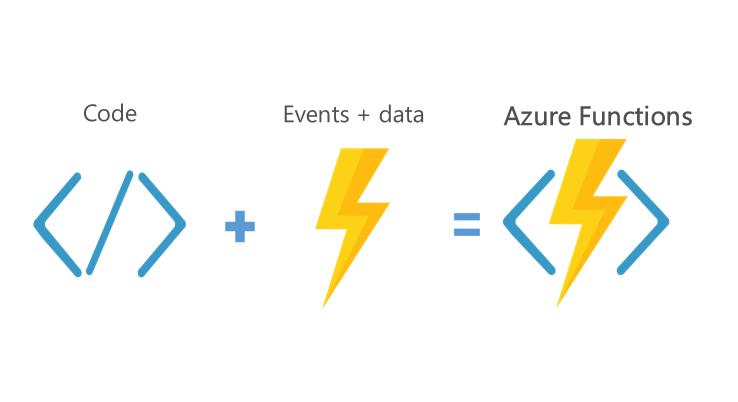
* 1. **File Server by Azure Function มีการทางานดังต่อไปนี้**
     1. เก็บไฟล์งาน เว็บ รูปภาพและอื่น ๆ ผ่าน Azure Function
     2. เรียกใช้งานไฟล์ที่ถูกเก็บ ตามที่ผู้ใช้งานต้องการ
     3. ไฟล์ที่ถูกจัดเก็บเป็นการทำงานในระบบ Cloud Server บน Microsoft Azure
     4. ทดสอบการใช้งาน File Server ดังกล่าว
     5. วิเคราะห์ผลการดาเนินงาน
     6. ปรับปรุงประสิทธิภาพ และความถูกต้อง
  2. **Function App Proxy สำหรับเรียกใช้งาน File Server Function App มีการทางานดังต่อไปนี้**
     1. **กำหนดเส้นทาง** path **ของ** URL **ในการเรียกใช้งาน** File Server Function App
     2. **กำหนดแหล่งข้อมูลที่ต้องการไปให้เป็นการทำงานของ** File Server Function App
     3. **ทดสอบการใช้งาน** Function App Proxy **ดังกล่าว**
     4. **วิเคราะห์ผลการดาเนินงาน**
     5. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  3. **ใช้ Azure Active Directory** ให้ทำงานร่วมกับ **File Server Function App**
     1. **ตรวจสอบตัวตนของของผู้ใช้งาน**
     2. **ยืนยันการเข้าใช้งานของ ผู้ใช้งานที่ถูกต้องให้ทำงาน** File Server Function App
     3. **ปฏิเสธการเข้าถึงของ ผู้ใช้งานที่ไม่อยู่ในกฎที่ตั้งไว้ของ** Azure Active Directory
     4. **ทดสอบผลลัพธ์ของ** Azure Active Directory **ดังกล่าว**
     5. **วิเคราะห์ผลการดาเนินงาน**
     6. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  4. ใช้ **Azure Content Delivery Network** ให้ทำงานร่วมกับ **File Server Function App**
     1. **กำหนดแหล่งที่มาของข้อมูลหลักให้เป็น** File Server Function App
     2. **กระจาย** Endpoint **ของแหล่งข้อมูลหลักออกไปใน** Cloud Server **บน** MicrosoftAzure
     3. **กำหนดชื่อ** Domain **ให้กับ** CDNEndpoint
     4. **ทดสอบการใช้งานของ** AzureContentDeliveryNetwork **ดังกล่าว**
     5. **วิเคราะห์ผลการดาเนินงาน**
     6. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  5. **Task Management Function** เรื่อง **Create Task in Trello Function** มีการทางานดังต่อไปนี้
     1. **รับค่าของข้อมมูลเข้ามาทำงานในฟังก์ชั่น**
     2. **นำข้อมูลที่ได้มาประมวลผลการทำงาน**
     3. **นำ Token และ Key ของ Trello มาเพื่อใช้งาน**
     4. **เชื่อมต่อกับ API ของ Trello เพื่อสามารถทำงานร่วมกันได้**
     5. **สร้าง Task บน Trello ตามข้อมูลที่ได้มา**
     6. **ทดสอบผลลัพธ์ของฟังก์ชัน**
     7. **วิเคราะห์ผลการดาเนินงาน**
     8. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  6. **Task Management Function** เรื่อง **Create Task in Microsoft To-Do** มีการทำงานดังต่อไปนี้
     1. **รับค่าของข้อมมูลเข้ามาทำงานในฟังก์ชัน**
     2. **นำข้อมูลที่ได้มาประมวลผลการทำงาน**
     3. **นำ** Token **และ** Key **ของ** MicrosoftTo-Do **มาเพื่อใช้งาน**
     4. **เชื่อมต่อกับ** API **ของ** MicrosoftTo-Do **เพื่อสามารถทำงานร่วมกันได้**
     5. **สร้าง** Task **บน** Microsoft To-Do **ตามข้อมูลที่ได้มา**
     6. **ทดสอบผลลัพธ์ของฟังก์ชัน**
     7. **วิเคราะห์ผลการดาเนินงาน**
     8. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  7. **Task Management Function** เรื่อง **Create Task in Microsoft Planner** มีการทำงานดังต่อไปนี้
     1. **รับค่าของข้อมมูลเข้ามาทำงานในฟังก์ชัน**
     2. **นำข้อมูลที่ได้มาประมวลผลการทำงาน**
     3. **นำ** Token **และ** Key **ของ** MicrosoftPlanner **มาเพื่อใช้งาน**
     4. **เชื่อมต่อกับ** API **ของ** MicrosoftPlanner **เพื่อสามารถทำงานร่วมกันได้**
     5. **สร้าง** Task **บน** MicrosoftPlanner **ตามข้อมูลที่ได้มา**
     6. **ทดสอบการใช้งานของฟังก์ชัน**
     7. **วิเคราะห์ผลการดาเนินงาน**
     8. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  8. **Function App Proxy** สำหรับเรียกใช้งาน **Task Management Function** มีการทำงานดังต่อไปนี้
     1. **กำหนดเส้นทาง** path **ของ** URL **ในการเรียกใช้งาน** TaskManagementFunction
     2. **กำหนดแหล่งข้อมูลที่ต้องการไปให้เป็นการทำงานของ** TaskManagementFunction
     3. **ทดสอบการใช้งาน** FunctionAppProxy **ดังกล่าว**
     4. **วิเคราะห์ผลการดาเนินงาน**
     5. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  9. **Logic App service สำหรับเรียกใช้งาน Task Management Function มีการทางานดังต่อไปนี้**
     1. **รับค่าของข้อมูลที่ผู้ใช้งานส่งเข้ามาเก็บไว้**
     2. **ส่งข้อมูลที่ได้ไปทำงานยัง** Task Management Function
     3. **รวบรวมการทำงานของ** Task Management **ทั้งสามฟังก์ชัน**
     4. **ทดสอบการใช้งานของ** Task Management Function **ทั้งสามว่าสามารถทำงานร่วมกันได้หรือไม่**
     5. **วิเคราะห์ผลการดาเนินงาน**
     6. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**
  10. **Create Task Management Form by Form.IO** มีการทำงานดังต่อไปนี้
      1. **สร้างฟอร์มในการรับข้อมูล จากผู้ใช้งาน**
      2. **แส้งหน้าฟอร์มผ่านไฟล์** HTML
      3. **ทดสอบผลลัพธ์ของหน้าฟอร์มในการแส้งผลผ่านไฟล์** HTML
      4. **วิเคราะห์ผลการดาเนินงาน**
      5. **ปรับปรุงประสิทธิภาพ และความถูกต้อง**

ประโยชน์ที่คาดว่าจะได้รับ

* + **แก้ปัญหาในเรื่องการลดค่าใช้จ่าย การใช้** App Service **แสดงผลเว็บขนาดเล็ก**
  + **บริษัทมี** File Server **ที่สามารถทำงานบน** Azure Function App
  + **สามารถดูแล จัดการ บริหารได้ด้วยตัวเอง**
  + **มีความสามารถในเรื่อง** Security **ในการเข้าถึงข้อมูลภายใน** File Server
  + **สามารถนำไปพัฒนาใช้กับองค์กร และระบบอื่น ๆ ได้**
  + **สามารถใช้ติดต่อสื่อสารงานกันได้โดยไม่ทำให้เสียสมาธิในการทำงาน ต่างจากการสื่อสารผ่านระบบ** Chat
  + **สามารถติดตามงานได้หลากหลายประเภทอย่างยืดหยุ่น รองรับการใช้งานได้ทุกแผนกในองค์กร**

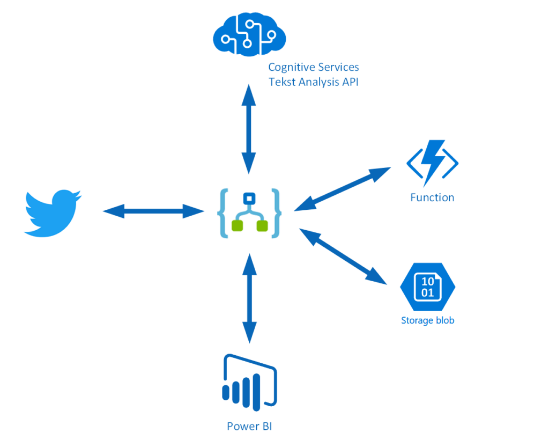
# **บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง**

Microsoft Azure Function App

**Microsoft Azure Function App นั้นคือการสร้าง Function ที่ทำงานบน Cloud Server โดยใน Function จะมีภาษาให้เลือกใช้ก็คือ C#, F#, Node.JS, Java และ PHP  
Azure Functions คือ รูปแบบการทำระบบ Back-end ในรูปแบบใหม่โดยที่ไม่จำเป็นต้องเปิดเครื่องทิ้งไว้เพื่อรอ Request หมายถึงไม่ต้องเสียเงินค่าเช่าเครื่อง server ตามระยะเวลา standby นั่นเอง แต่การคิดค่าใช้จ่ายจะเกิดขึ้นตอนเกิด event หรือเกิด Request รูปแบบนี้เราเรียกันว่า Serverless Architecture**

ภาพที่ 1ตัวอย่างการกระบวนการของ Function App

Azure logic app

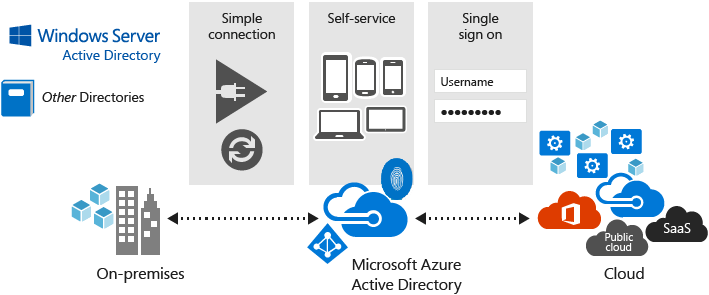
**Logic apps คือหนึ่งในserviceของazureที่ใช้เชื่อมต่อการทำงานของแอปพลิเคชัน ข้อมูลและอุปกรณ์ต่าง ๆ เข้าด้วยกันผ่านระบบคลาวด์ได้อย่างง่ายดายบนระบบซอฟต์แวร์ขนาดใหญ่ของ Microsoftตัวอย่างการแอปพลิเคชันต่าง ๆ ที่สามารถนำมาเชื่อมต่อกันได้เช่น Salesforce, Office 365, Twitter, Dropbox, Google servicesและอื่น ๆ**

ภาพที่ 2 ตัวอย่างการกระบวนการของ Logic App

Azure Active Directory

**Azure Active Directory หรือ Azure AD เป็นบริการ Identity & Access Management สำหรับควบคุมการเข้าถึงระบบ Cloud ซึ่งช่วยให้องค์กรสามารถทำ Single Sign-on ระหว่างระบบใน Data Center และระบบ Cloud ได้อย่างไร้รอยต่อ นั่นหมายความว่า หลังจากที่พนักงานในองค์กรทำการพิสูจน์ตัวตนเข้าสู่ระบบผ่าน AD ใน Data Center แล้ว จะสามารถเข้าใช้งานแอปพลิเคชันต่าง ๆ บนระบบ Cloud ไม่ว่าจะเป็น Office 365, Dynamic CRM Online, SalesForce.com หรือ Dropbox ได้ทันที โดยไม่ต้องพิสูจน์ตัวตนซ้ำ**

**คุณสมบัติเด่นของ Azure AD ได้แก่**

* **รองรับการพิสูจน์ตัวตนแบบ Multi-factor Authentication เสริมความแข็งแกร่งในการตรวจสอบผู้ใช้งานก่อนเข้าถึงแอปพลิเคชัน**
* **Self-service Password Management และ Self Service Group Management สำหรับให้ผู้ใช้สามารถรีเซ็ตรหัสผ่านและบริหารจัดการกลุ่มของตนได้ด้วยตัวเอง**
* **รองรับการทำงานร่วมกับ Cloud Applications ที่พัฒนาขึ้นมาเอง เพื่อให้จัดการเรื่อง SSO และสิทธิ์ในการใช้งานได้ผ่านทาง SAML 2.0, WS-\* Protocol, OpenID และ OAuth**
* **ให้บริการภายใต้โครงข่ายมาตรฐานสูงของ Microsoft โดยรองรับ SLA ที่ 99.9%**

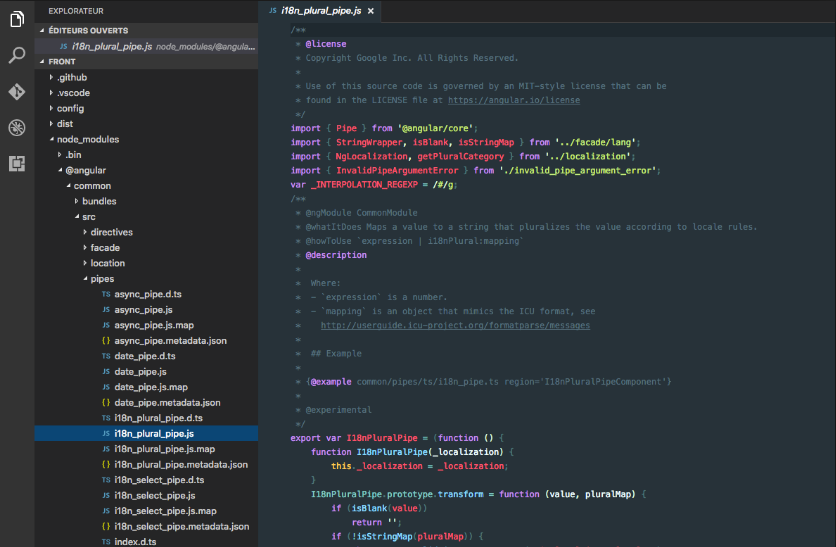
ภาพที่ 3 ตัวอย่างการกระบวนการของ Microsoft Azure Active Directory

Azure Content Delivery Network

**CDN (Content Delivery Network) คือระบบเครือข่ายคอมพิวเตอร์ขนาดใหญ่ กระจายไปตามภูมิภาคต่าง ๆ ทั่วโลก โดยที่ระบบคอมพิวเตอร์เหล่านี้จะเชื่อมต่อกับอินเทอร์เน็ตความเร็วสูง เพื่อทำหน้าที่ในการส่งข้อมูล ไปถึงผู้รับให้เร็วที่สุด โดยใช้เครือข่ายที่ใกล้กับผู้รับมากที่สุด หากเว็บไซต์ของเราเป็นเว็บเล็ก ๆ มีกลุ่มเป้าหมายเป็นผู้คนในประเทศ คนเข้าใช้งานไม่สูงมาก CDN ก็อาจจะไม่จำเป็นสำหรับเว็บของคุณ แต่หากเว็บไซต์ของคุณเป็นเว็บขนาดใหญ่ มีการโหลดข้อมูลในปริมาณมหาศาล ทั้งยังมีกลุ่มเป้าหมายเป็นผู้คนทั่วทั้งโลก CDN นี้แหละคือพระเอก ที่ช่วยเสริมเขี้ยวเล็บ เพิ่มประสิทธิภาพให้เว็บของคุณ**

ภาพที่ 4 ตัวอย่างการกระบวนการของ Microsoft Content Delivery Network

Visual Studio Code

**Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor โปรแกรมหนึ่งที่ใช้ในการในการเขียนโค้ดโปรแกรม ช่วยลดระยะเวลาการเขียนโค้ดให้น้อยลงทำงานได้ง่ายขึ้น ลักษณะที่สำคัญอย่างหนึ่งของการเขียนโปรแกรมบน Visual Studio Code คือ ติดตั้งง่าย โปรแกรมมีขนาดเล็ก มีฟังก์ชั่นในการ จัดรูปแบบโค้ด Plugin และ Theme มากมายที่สามารถลงเพิ่มเติมได้ในภายหลัง Visual Studio Code เป็นเครื่องมือที่ช่วยให้การเขียนโปรแกรมของโปรแกรมเมอร์ให้สะดวกรวดเร็วมากขึ้น ในส่วนภาษาโปรแกรมที่ตัว Visual Studio Code รองรับก็มีมาให้มากมายยกตัวอย่าง เช่น HTML CSS JAVA และ PHP เป็นต้น โปรแกรม Visual Studio Code ยังเปิดโอกาสให้นักพัฒนาจัดทำ Theme ของ โปรแกรม Visual Studio Code ขึ้นมาช่วยให้ตัวโปรแกรมมีความสวยงามมากขึ้นใช้งานได้ง่ายขึ้นปรับแต่ง ให้เข้ากับการทำงานของแต่ละคนได้อย่างหลากหลาย ช่วยปรับปรุงให้โค้ดโปรแกรมต่าง ๆ มีความเป็นระเบียบมีสีสันสวยงาม ทำให้คนในทีมอ่านและเข้าใจโค้ดของสมาชิกคนอื่น ๆ ได้ง่ายขึ้น นอกจากนี้โปรแกรม Visual Studio Code ยังช่วยประหยัดเวลาในการสร้างโค้ดโปรแกรมได้ถึงหนึ่งเท่าตัว ช่วยในการทำงานการเขียนโค้ดโปรแกรมของทางบริษัทได้เป็นอย่างดีเพิ่มความสะดวกแทนที่จะมานั่งพิมพ์โค้ดโปรแกรมทีละตัวในโปรแกรม Visual Studio Code ก็จะมีระบบค้นหาคำใกล้เคียงให้กับโปรแกรมเมอร์ช่วยให้ลดระยะเวลาไปได้มาก มีมินิแม็พของโค้ดทำให้สามารถไปยังบริเวณที่จะแก้ไข โค้ดได้รวดเร็วยิ่งขึ้น และที่สำคัญโปรแกรม Sublime text สามารถนำมาติดตั้งและใช้งานได้ฟรีร่วมถึง Theme Plugin และฟังก์ชั่นอื่น ๆ ส่วนใหญ่ก็สามารถหาและติดตั้งได้ฟรีเช่นเดียวกัน จากที่กล่าวมาจะเห็นได้ว่าโปรแกรม Visual Studio Code มีประสิทธิภาพและความคุ้มค่าอย่างมาก ที่ให้ผู้ใช้สามารถนำมาพัฒนาโค้ดโปรแกรมด้วยความสะดวกยิ่งขึ้น นำมาใช้ทำงานในการสร้างสรรค์ ผลงานโค้ดโปรแกรม หรือเว็บไซต์ต่าง ๆ ได้เนื่องจากรองรับภาษาโปรแกรมมากมาย ดังนั้นการใช้ Visual Studio Code ที่สามารถผสมผสานภาษาในการเขียนโค้ดโปรแกรมร่วมกับ Framework อื่น ๆ ได้นั้น ทำให้องค์กรมีค่าใช้จ่ายที่ลดลงแต่การทำงานยังคงมี****ประสิทธิภาพ**

ภาพที่ 5 ภาพตัวอย่างของโปรแกรม Visual Studio Code

ภาพที่ 6 ภาพสัญลักษณ์ของโปรแกรม Visual Studio Code

C#

**C# คือ ภาษาของคอมพิวเตอร์ประเภท object oriented programming ซึ่งถูกพัฒนาโดย Microsoft เพื่อมีจุดมุ่งหมาย ในการทำการรวมความสามารถของการคำนวณของC++ ด้วยการใช้โปรแกรมที่ง่ายกว่าของ visual basic โดย C# มีพื้นฐานจากC++และเก็บส่วนของการทำงานคล้ายกับจาวา ซึ่งสำหรับ C#ได้รับการอกแบให้ทำงานกับNet platformของ Microsoft เพื่อมีจุดมุ่งหมายอำนวยความสะดวกในการแลกเปลี่ยนการสารสนเทศและบริการต่างๆผ่านเว็บไซต์ซึ่งจะทำให้ผู้พัฒนา สามารถที่จะสร้างโปรแกรมประยุกต์ในขนาดที่กะทัดรัด และนอกจากนี้ยังทำให้โปรแกรมง่ายขึ้นเมื่อผ่านการใช้ Extensible markup languageและ Simple Object Access Protocol ซึ่งยอมให้เข้าถึง object ของโปรแกรมหรือเมธอด โดยปราศจากเรื่องของความต้องการที่จะให้ผู้เขียนเขียนโปรแกรมคำสั่งเพิ่มเติมในแต่ละขั้นตอน เพราะเนื่องจากว่าผู้เขียนโปรแกรมนั้นสามารถสร้างบนคำสั่งที่มีอยู่ก่อนหน้านี้ได้แล้ว ซึ่งใช้แทนที่ของการลอกซ้ำ และนอกจากนี้ยังถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ net frameworkโดยถ้าจะให้พูดอย่าง่ายๆนั้นภาษา C#มีความใกล้เคียงกับภาษาจาวามากที่สุด เพราะฉะนั้นสำหรับนักเขียนโปรแกรมภาษาจาวาจึงสามารถที่จะย้ายมาเขียนภาษา C#ได้โดยง่าย และนอกจากนี้ก็ยังเป็นแบบสืบจากคลาสหลักได้คลาสเดียว เพราะสำหรับจุดเด่นที่สำคัญของภาษานี้คือเป็นภาษาที่เน้นชิ้นส่วนถูกออกแบมาเป็นอย่างดีเพื่อที่จะทำให้สามารถนำมาใช้ต่อการเป็นอะไรก็ได้และนอกจากนี้ สิ่งต่าง ๆ ในภาษา C#ก็ถือเป็น object ทั้งหมด**

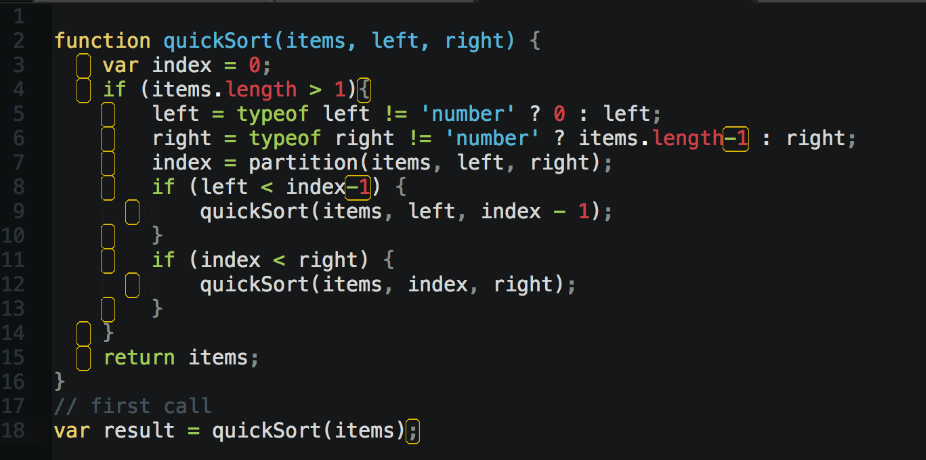
ภาพที่ 7ภาพสัญลักษณ์ของ C#

JavaScript

**JavaScript คือภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และทางฝั่งเซิร์ฟเวอร์ (Server)**

**JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript สามารถทำให้ การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือการกรอกข้อความในฟอร์ม เป็นต้น**

**เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิดที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวางรวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น Client-Side Script) ดังนั้น JavaScript จึงสามารถทำงานได้เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ดี สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชั่นใหม่ ๆ ออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนเบราว์เซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด Error ได้**



ภาพที่ 8 ภาพตัวอย่างของ JavaScript

ภาพที่ 9 ภาพสัญลักษณ์ของ JavaScript

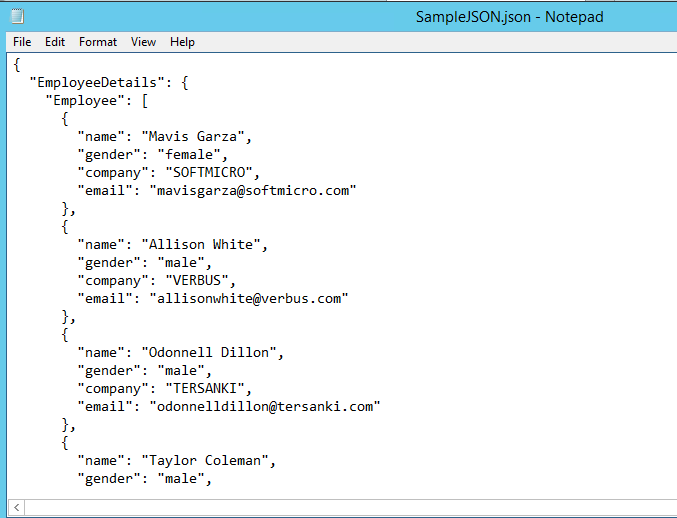
Json

**JSON (JavaScript Object Notation) รูปแบบการรับส่งข้อมูลคอมพิเตอร์ขนาดเล็ก เป็นภาษาที่อ่าน, เขียน และสร้างเป็นภาษาโปรแกรมที่เข้าใจง่าย ถูกสร้างขึ้นมาโดย ดักกลาส คลอฟอร์ด ต้นกำเนิดอยู่ในรูปแบบ ของ RFC 4627 and ECMA-404**

**JSON เป็นรูปแบบข้อความภาษาอิสระ แต่ตัวภาษาจะใช้เป็นภาษาทางการเหมือนกับ ภาษา C รวมถึงC, C++, C#, Java, JavaScript, Perl, Python และ อื่นๆ ส่วนใหญ่่ JSON จะนิยมใช้มากกว่า XML เพราะเข้าใจง่ายกว่า ไม่ซับซ้อน ทำให้ JSON รูปแบบการรับส่งข้อมูลที่ยอดเยี่ยม**

**JSON ถูกสร้าง 2 รูปแบบ**

* **ถ้าเป็นรูปแบบของชื่อหรือข้อมูลคู่กัน จะบันทึกมาเป็นรูปแบบของ Object, การบันทึก , ประโยค, คำศัพท์, รายชื่อ, โครงสร้างแบบแฮช หรือ associative array**
* **ถ้าเป็นรูปแบบของรายชื่อข้อมูล โดยทั่วไป จะบันทึกมาเป็นรูปแบบของ array, vector, list หรือ**

API

ภาพที่ 10 ภาพตัวอย่างข้อมูล Json

**API (Application Programming Interface) คือช่องทางการเชื่อมต่อ ช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทางานเข้ากับระบบปฏิบัติการ**

**ตัวอย่างเช่น Twitter มีหลายเว็บ ที่มีการเชื่อมโยงข้อมูลกับ Twitter ทั้งเป็นการอ่านข้อมูลจาก twitter หรือ ส่งข้อมูลเข้า Twitter เองก็ตาม ซึ่งล้วนอาศัยการเชื่อมต่อแลกเปลี่ยนข้อมูลกันด้วย API นั้นเอง และอีกหนึ่งตัวอย่าง เช่น Google Maps API คือบริการของ Google อีกรูปแบบหนึ่งที่เราสามารถนำข้อมูลของ Google Maps ที่ทาง Google ให้บริการโดยส่วนมากจะนำมาใช้กับเว็บไซต์ของบริษัทฯ หรือเว็บไซต์ห้างร้านต่าง ๆ เพื่อเป็นอีกช่องทางที่ให้ลูกค้ารู้ว่าบริษัทฯ หรือห้างร้านนั้นตั้งอยู่ที่**

ภาพที่ 11 ภาพตัวอย่างของ API

# บทที่ 3 ขั้นตอนการพัฒนาและการออกแบบ

...................

1. ศึกษาการเลือกใช้โปรแกรมให้เหมาะสมกับระบบงาน
   1. การเลือกใช้โปรแกรมให้เหมาะสมกับระบบงาน.........
2. ขั้นตอนออกแบบและพัฒนาระบบ
   1. วางแผนออกแบบฟังก์ชันและหน้าเวบไซต์

..........อธิบาย......

* + 1. ออกแบบฟังก์ชัน
       1. ฟังก์ชัน FileServer
       2. ฟังก์ชัน Trello
    2. ออกแบบหน้าเว็บไซต์

.......อธิบาย...........

* 1. ผลการออกแบบฟังก์ชันและหน้าเว็บไซต์
     1. ส่วนของ FileServer
        1. ส่วนของการสร้าง Function App สำหรับ FileServer
        2. ส่วนของการสร้างฟังก์ชัน FileServer
        3. ส่วนของการทำ Active Directory ให้ฟังก์ชัน FileServer
        4. ส่วนของการทำ Content Network Delivery ให้ฟังก์ชัน FileServer
     2. ฟหก
     3. ฟหก
     4. ฟหก
     5. ฟหก
     6. ฟหก
     7. ฟหกฟหกฟหก

................

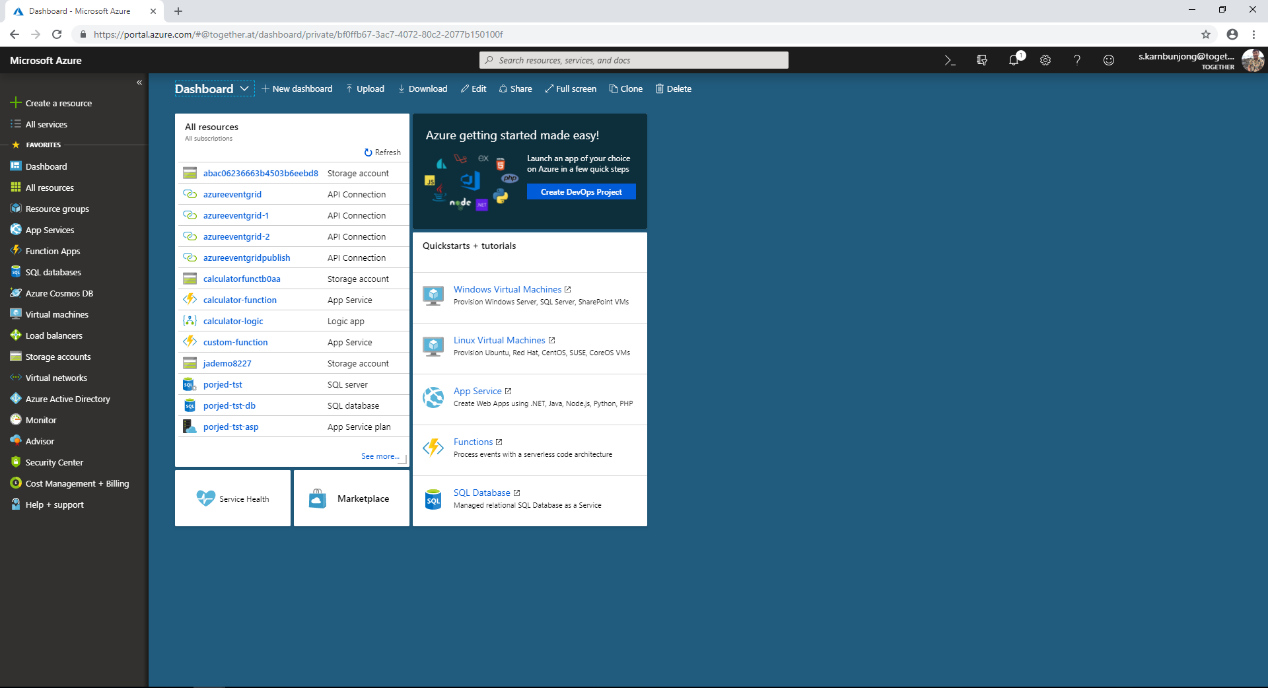
1. ศึกษาการใช้งาน Microsoft Azure Function App

เป็นเซอร์วิสบน Microsoft Azure เพื่อใช้ในการทำงานสร้างแอพพลิเคชันของทาง Back-end

พัฒนาซอฟต์แวร์และการทำงานอื่นๆในรูปแบบของการเขียนคำสั่งในฟังก์ชันจะสามารถใช้ภาษา C#, F# ,NodeJS, Java Script, และ PHP ในการพัฒนา มีลักษณะพิเศษคือเป็นการทำงานผ่านระบบคลาวด์เซิฟเวอร์

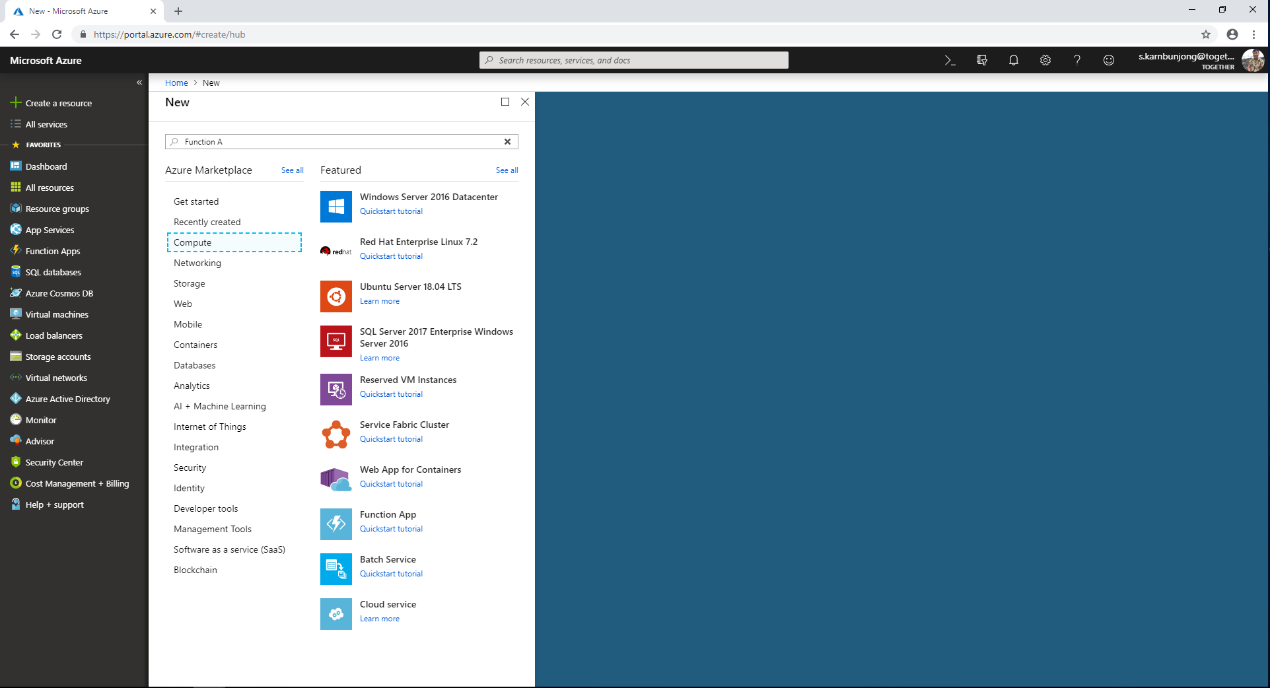
* 1. หน้าจอเริ่มต้นเมื่อเข้าใช้งาน Microsoft Azure ผ่านเว็บเบราว์เซอร์

ในหน้าหลักจะมีแถบทางด้สนซ้ายเซอร์วิสหลักแสดงอยู่ และมีตาราง All resources ที่คอยแสดงผลเซอร์วิสที่แอคเค้าผู้ใช้งานในขณะนั้นได้ทำการสร้างขึ้นมาเพื่อใช้งาน



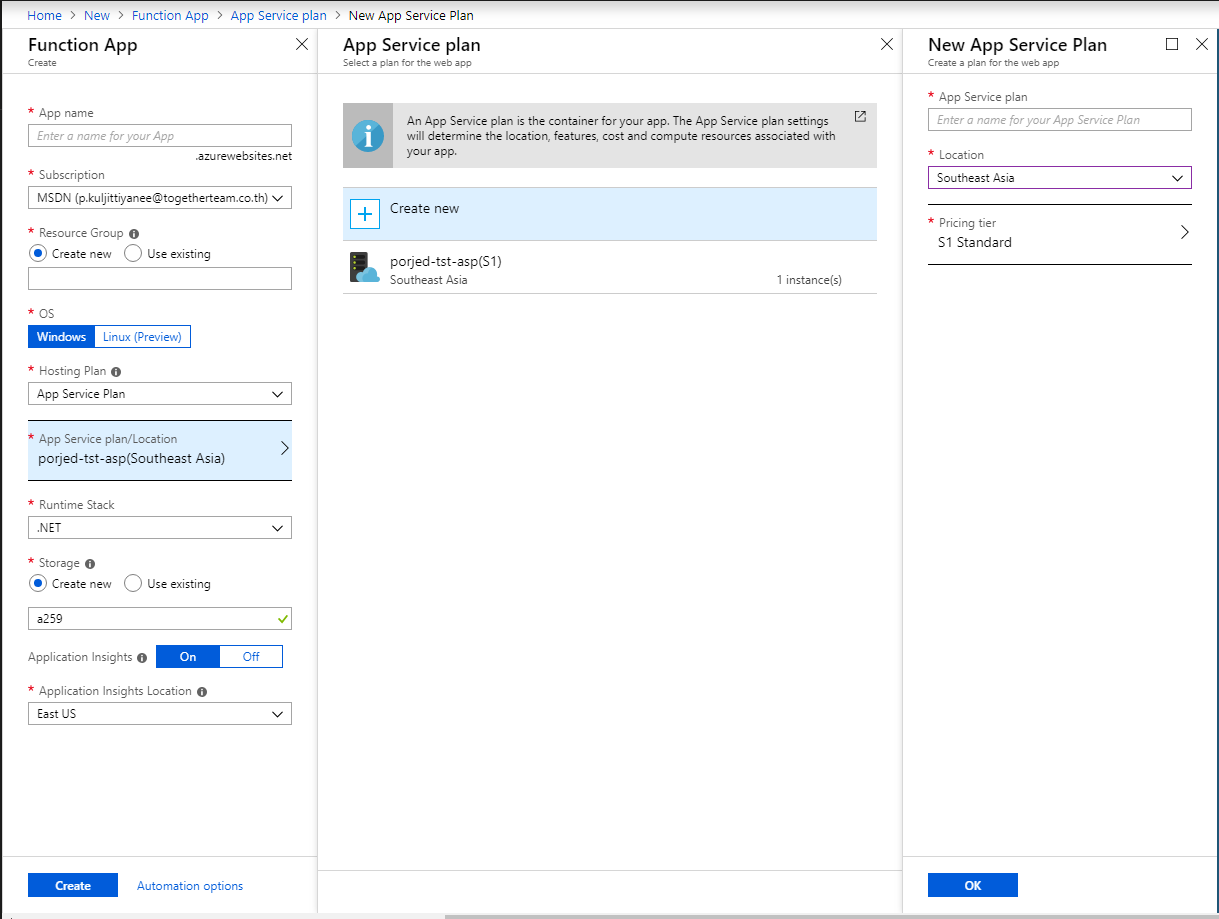
ภาพที่ 12 หน้าเริ่มต้น Microsoft Azure

* 1. การสร้าง Function App ในการทำ FileServer และ AddTaskManagement
  + แถบทางซ้ายมือเลือก Create a resource
  + ดูที่หมวด Compute ให้เลือก Feature ที่ชื่อ Function App
  + หรือจะค้นหาในช่องด้วยคำว่า Function App



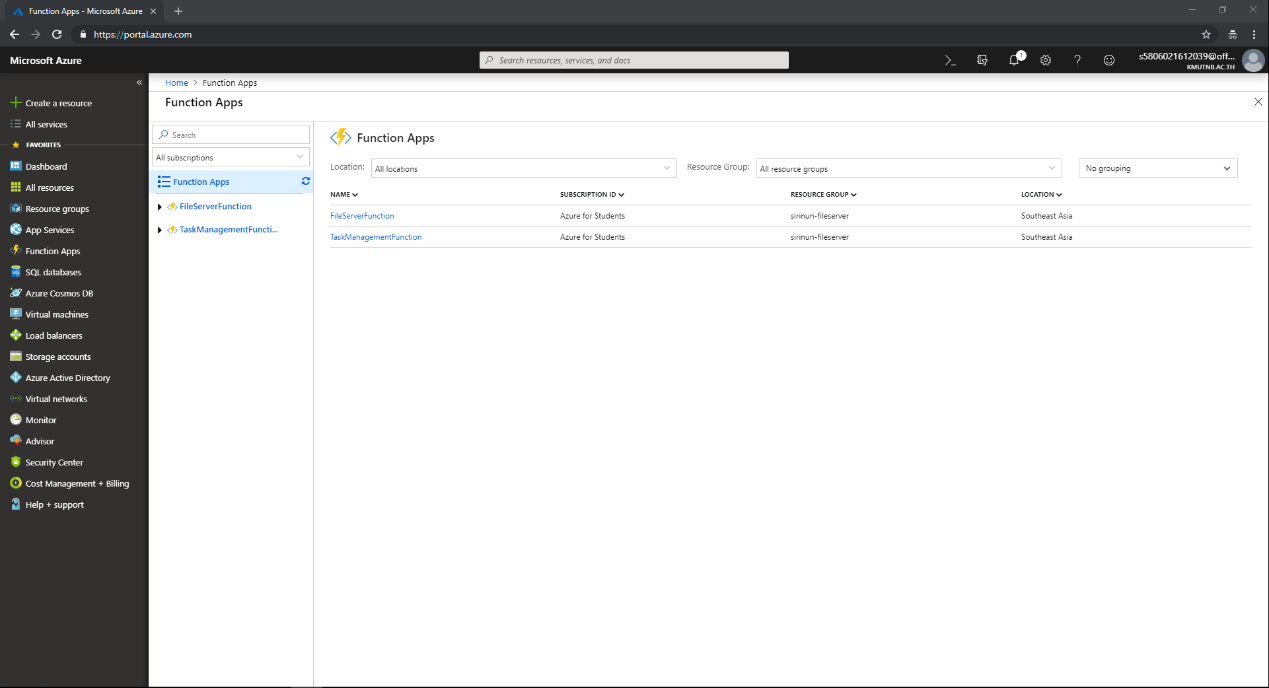
ภาพที่ 13 หน้าต่าง create a resource

* + เมื่ออยู่ในหน้า Create Function App แล้วให้ทำการกรอกข้อมูลให้เรียบร้อยได้แก่
    - App name เป็นชื่อของฟังก์ชันแอพที่ต้องการ
    - Subscription เป็นช่องทางสถานะทางการเงินสำหรับชำระค่าบริการ
    - Resource Group เป็นกลุ่มสำหรับแบ่งการทำงานให้มาอยู่ร่วมกัน
    - OS เป็นระบบปฏิบัติการที่ใช้
    - Hosting Plan เป็นโหมดการทำงานของฟังก์ชันมีให้เลือก 2 ประเภทคือ  
      Consumption Plan และ App Service Plan โดยจะเลือกเป็น   
      App Service Plan แล้วเซ็ต location ให้เป็น Southeast Asia



ภาพที่ 14 หน้าต่างการตั้งค่า Hosting Plan

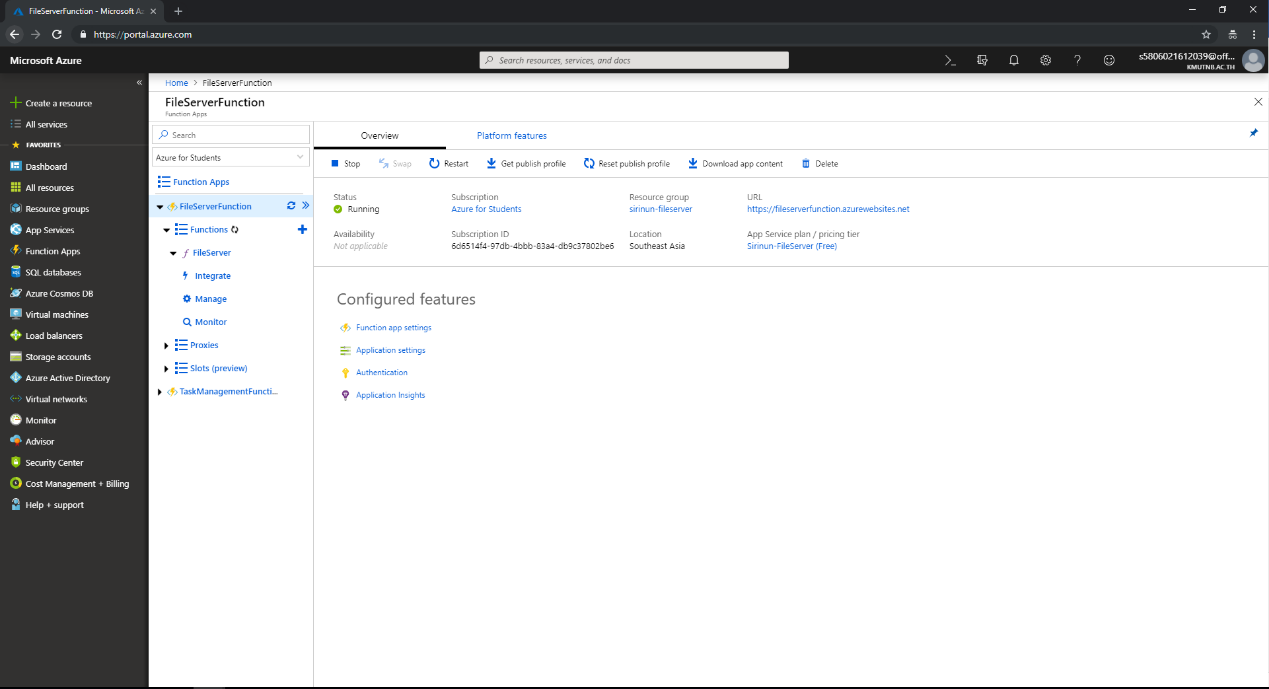
* + - Runtime Stack เลือกเป็น .NET
    - Storage เป็นแหล่งเก็บข้อมูลควรจะใช้ Storage อันเดียวกันทั้งหมดกับการสร้างเซอร์วิสต่างๆในงานนี้
    - Application Insights Location เลือกเป็น Southeast Asia
    - เมื่อกรอกข้อมูลครบหมดแล้วกดปุ่ม Create
  + เมื่อสร้าง Function App ของ File Server และ Task Management เสร็จแล้วให้ไปดูที่หน้าเริ่มต้น Microsoft Azure ในแถบทางขวาเลือก Function Apps จะมีหน้าต่างแสดงผล Function App ทั้งหมดที่สร้างขึ้นมา



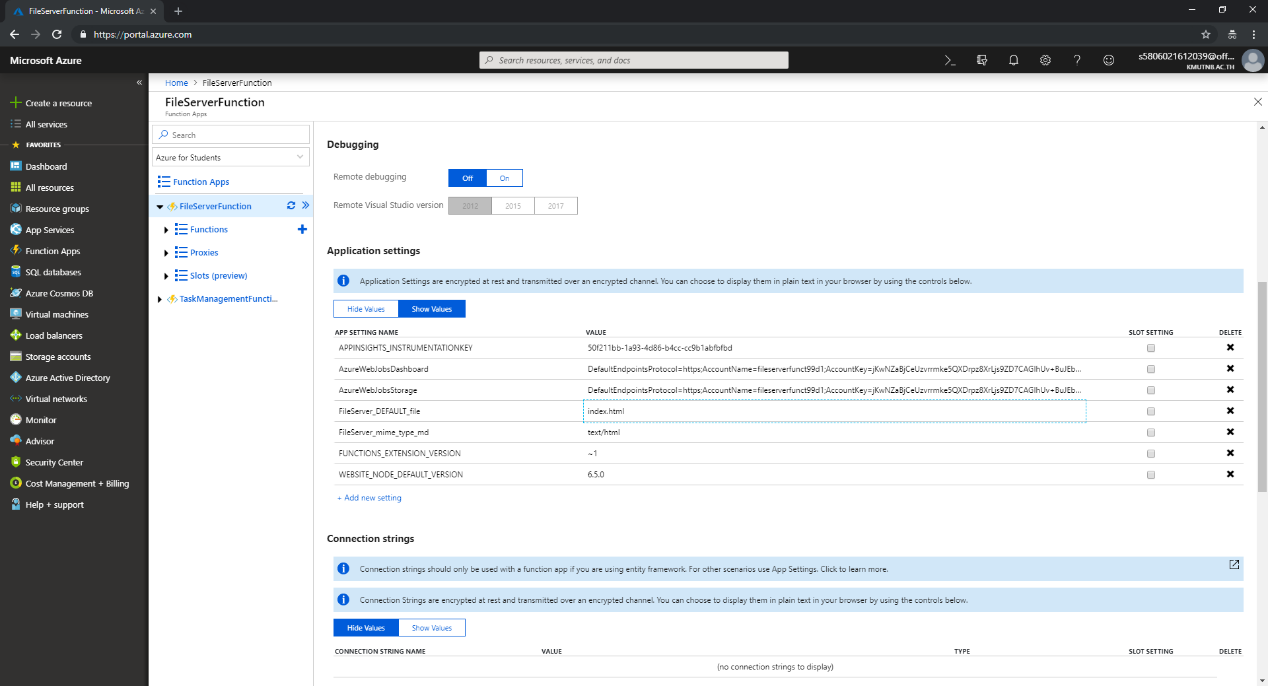
ภาพที่ 15 หน้าต่างแสดงผล Function App ทั้งหมด

จากรูปจะเห็นได้ว่ามี Function App อยู่ 2 ตัวที่ชื่อ FileServerFunction และ TaskManagementFunction

* 1. ในการทำฟังก์ชัน FileServer
  + เข้ามาที่ Function Apps ที่ชื่อ FileServerFunction ที่สร้างรอไว้
  + เลือก Overview ไปที่ Application settings คลิก Add new setting  
    ในช่อง APP SETTING NAME ใส่คำว่า “FileServer\_DEFAULT\_file”  
    ในช่อง VALUE ใส่คำว่า “index.html”



ภาพที่ 16 หน้าต่าง Overview ของ Function App



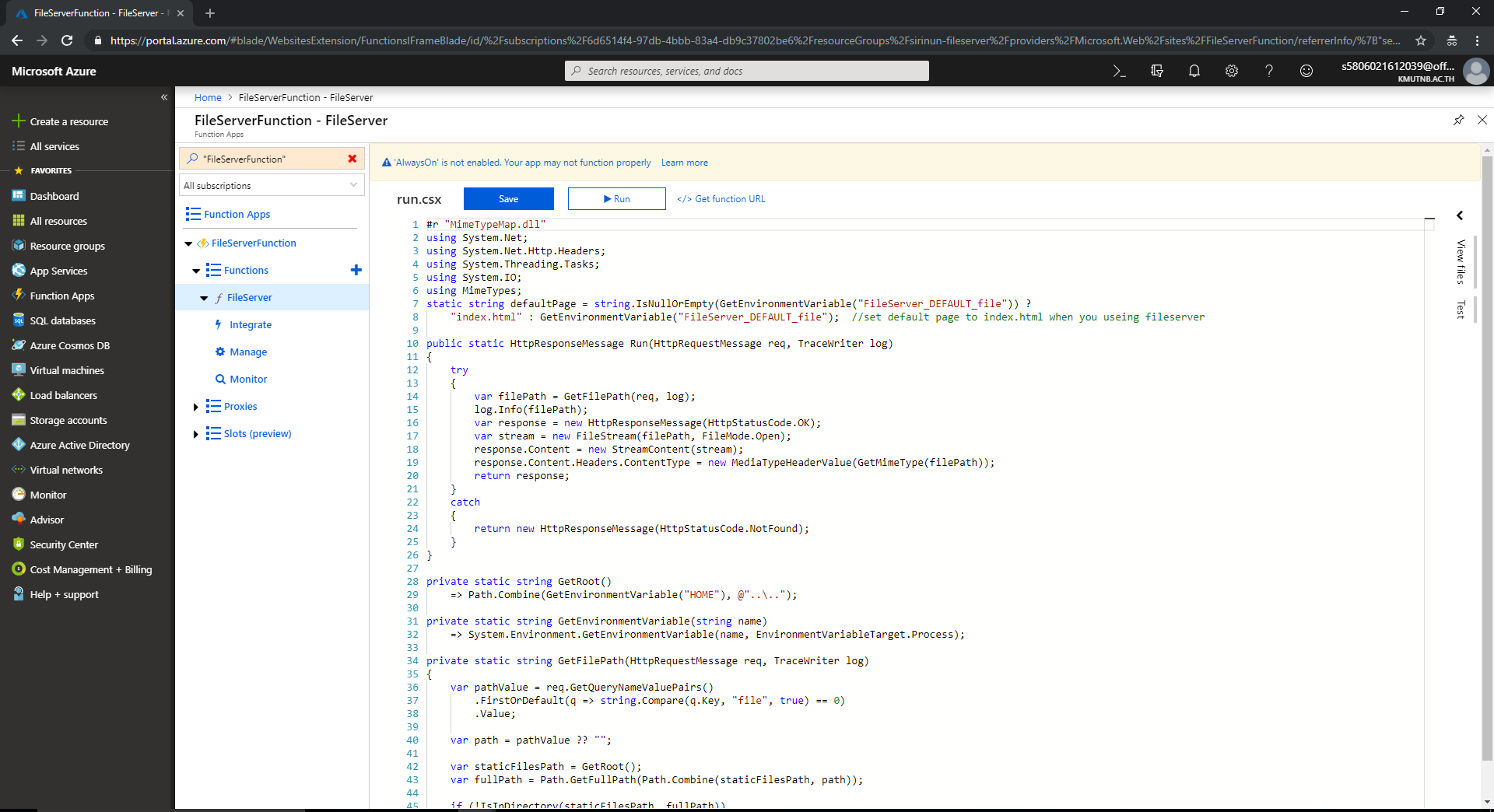
ภาพที่ 17 หน้าต่าง Application settings

* + เมื่อสร้าง Function App สังเกตุว่าในหน้า Overview จะมี URL ให้ไว้สำหรับผู้ใช้งานได้รีเควสเข้ามา **https://fileserverfunction.azurewebsites.net/**
  + เลือก Functions แล้วกด create new แล้วเลือกฟังก์ชัน C# แบบ HTTP trigger



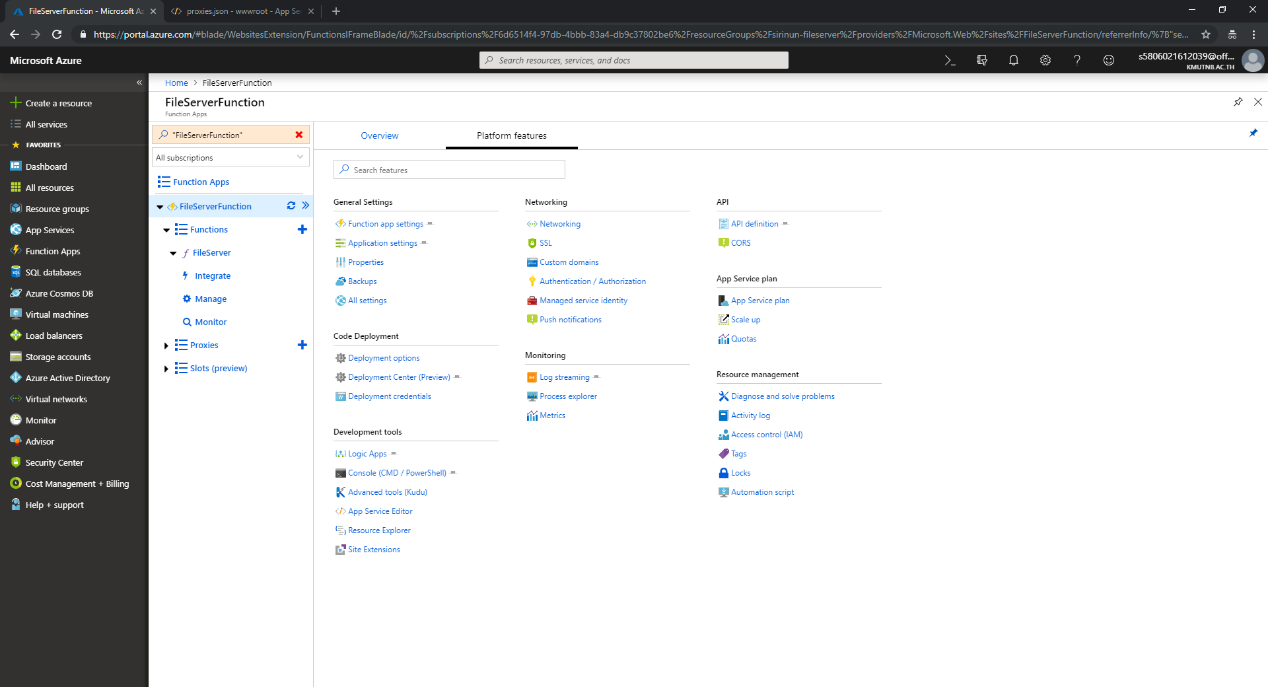
ภาพที่ 18 หน้าต่างสร้างฟังก์ชัน FileServer

* + หลังจากสร้างเสร็จแล้วก็เป็นการเขียนโค้ดคำสั่งให้ทำงาน



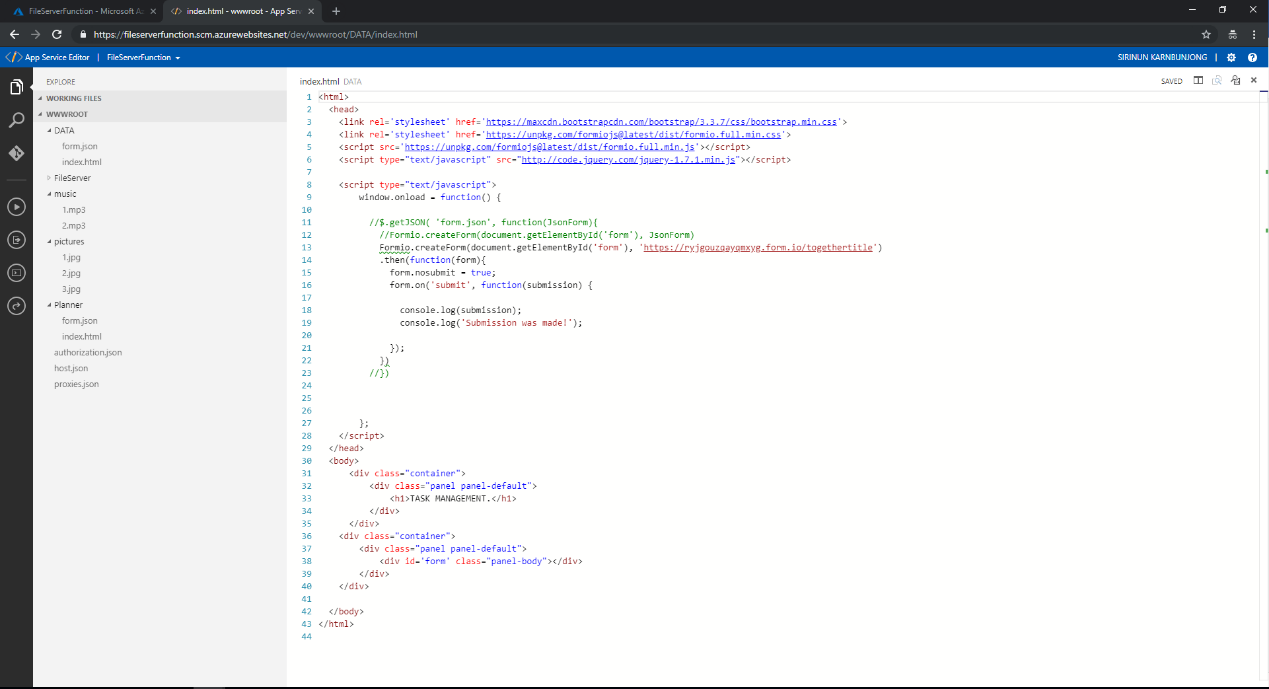
ภาพที่ 19 หน้าต่างฟังก์ชัน FileServer สำหรับเขียนคำสั่งการทำงาน

* 1. เก็บไฟล์ที่ต้องการเข้าไปใน
  + เข้าไปที่ Function Apps ที่ชื่อ FileServerFunction
  + เลือก Planform Feature ให้หาคำว่า App Service Editor



ภาพที่ 20 หน้าต่าง Platform features ของ Function App

* + เมื่อเปิด App Service Editor ขึ้นมาแล้วให้ดูแถบทางซ้ายเลือก Explore แล้วทำการสร้างโฟลเดอร์เพื่อใส่ไฟล์ที่ต้องการ ยกตัวอย่างเช่น สร้างโฟลเดอร์ชื่อ DATA แล้วอัพโหลดไฟล์ index.html เข้าไปเพื่อเก็บไว้รอเรียกใช้งาน

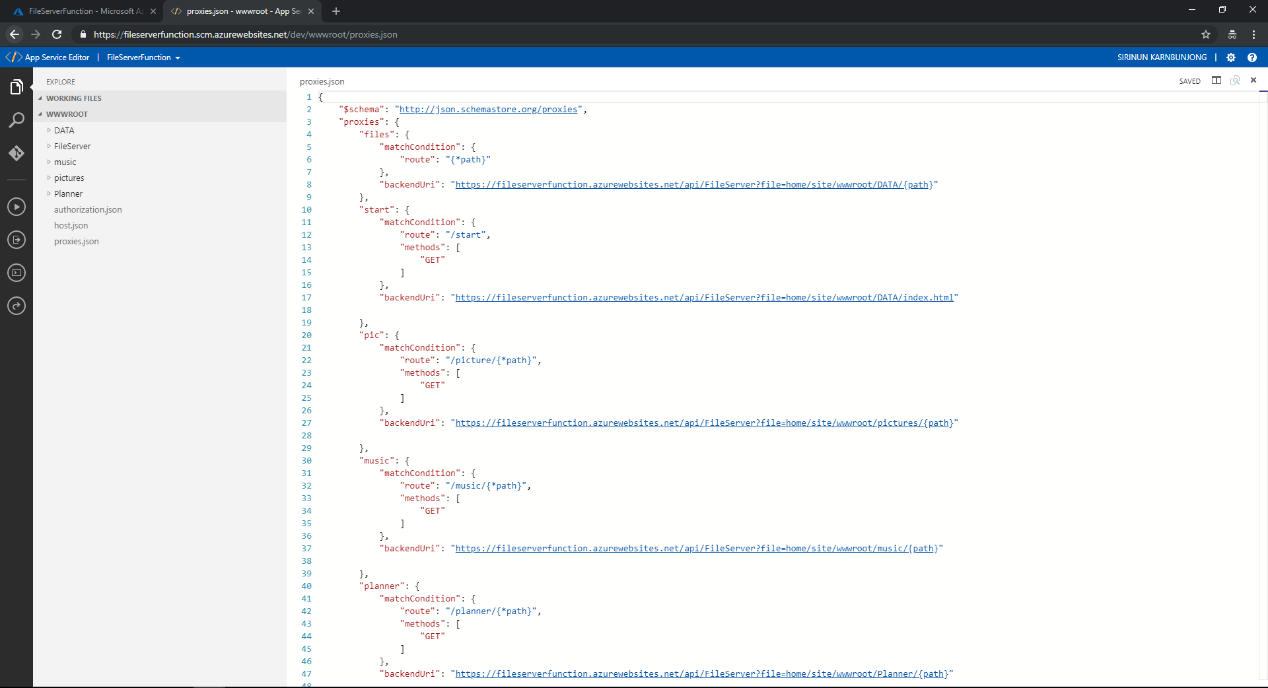


ภาพที่ 21 หน้าต่าง App Service Editor ที่ทำการเก็บไฟล์

* 1. ในการทำฟังก์ชัน Proxies

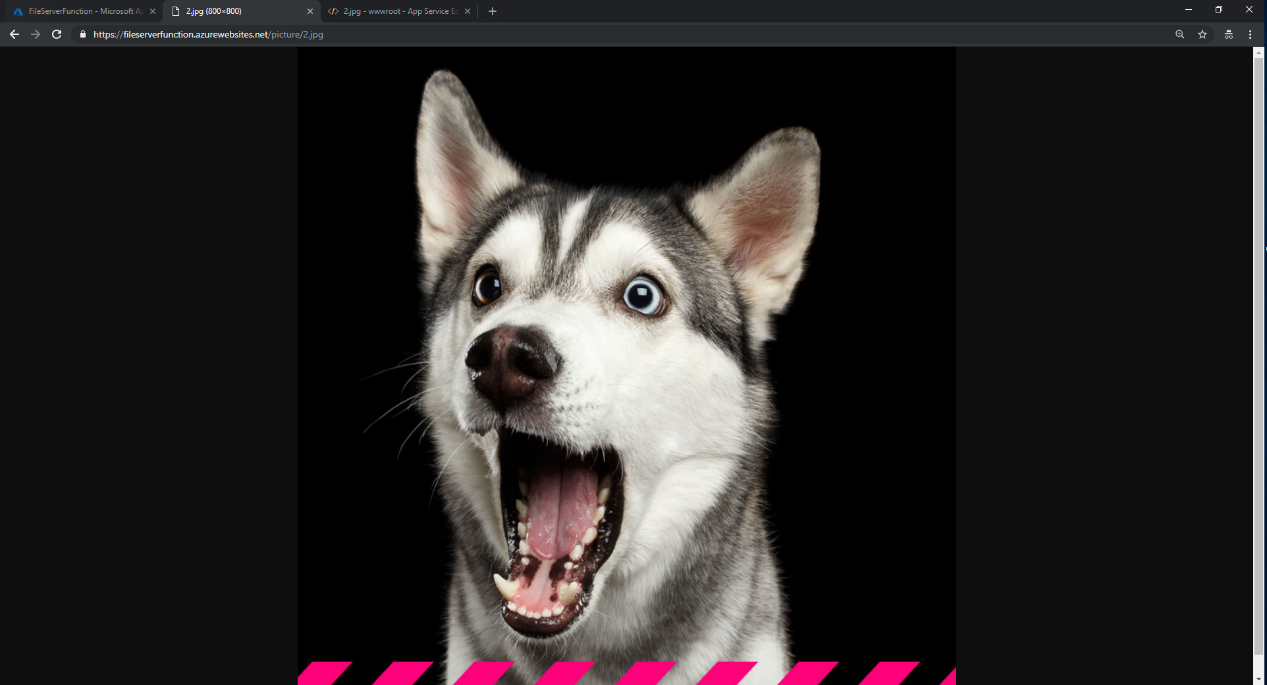
ในการเรียกใช้งานไฟล์จะมี Proxies เป็นตัวกำหนดเส้นทางที่ได้รับและเส้นทางจุดหมาย  
นั้นคือหากทำการเก็บไฟล์ที่ต้องการเรียกใช้ไว้ที่พาธ **“/wwwroot/Data”** (ภาพที่ 21)  
แล้วผู้ใช้งานรีเควสมาที่ **https://%WEBSITE\_SITE\_NAME%.azurewebsites.net/api/File-Server-v1?file=home/site/wwwrooot/{path}** แล้ว Proxies ก็จะเช็คว่าพาธที่ได้รับมาตรงกับจุดหมายปลายทางที่ไฟล์ไหนก็จะนำไฟล์นั้นออกมาแสดงผล

* + เข้ามาที่ Function Apps ที่ชื่อ FileServerFunction
  + เลือก Planform Feature ให้หาคำว่า App Service Editor
  + เมื่ออยู่ในหน้า App Service Editor แล้วให้เลือกหัวข้อ Explore ทางแถบซ้าย
  + คลิกขวาเพื่อทำการ new file ขึ้นมาตั้งชื่อเป็น proxies.json เพื่อทำการเซ็ตค่าของพาธที่ได้รับให้ไปยังปลายทางที่ไฟล์ใด



ภาพที่ 22 หน้าต่างแสดงไฟล์ proxies.json เพื่อเซ็ตค่า

* 1. ทดสอบการใช้งานของ FileServerFunction หลังจากเซ็ต proxies
* ในการทกสอบการใช้งานนั้นเริ่มลองจากเรียกใช้งานไฟล์รูปภาพที่ทำการเก็บไว้ยังพาธ**“/wwwroot/pictures/2.jpg”** (ภาพที่21)
* โดยเรียกใช้งานผ่านลิงค์ URL ของ Function App FileServerFunction   
  ดูได้ในหน้าของ Overview ของ Function Apps (ภาพที่ 16)  
  [**https://fileserverfunction.azurewebsites.net**](https://fileserverfunction.azurewebsites.net)
* ผู้ใช้งานจะเรียกใช้งานผ่าน URL โดยต่อด้วยพาธตามที่มีการเซ็ตไว้ใน proxies  
  ตัวอย่างคือจะเรียกไฟล์รูปภาพ 2.jpg (ภาพที่ 22) ซึ่งจะเรียกผ่าน URl  
  [**https://fileserverfunction.azurewebsites.net/picture/2.jpg**](https://fileserverfunction.azurewebsites.net/picture/2.jpg)



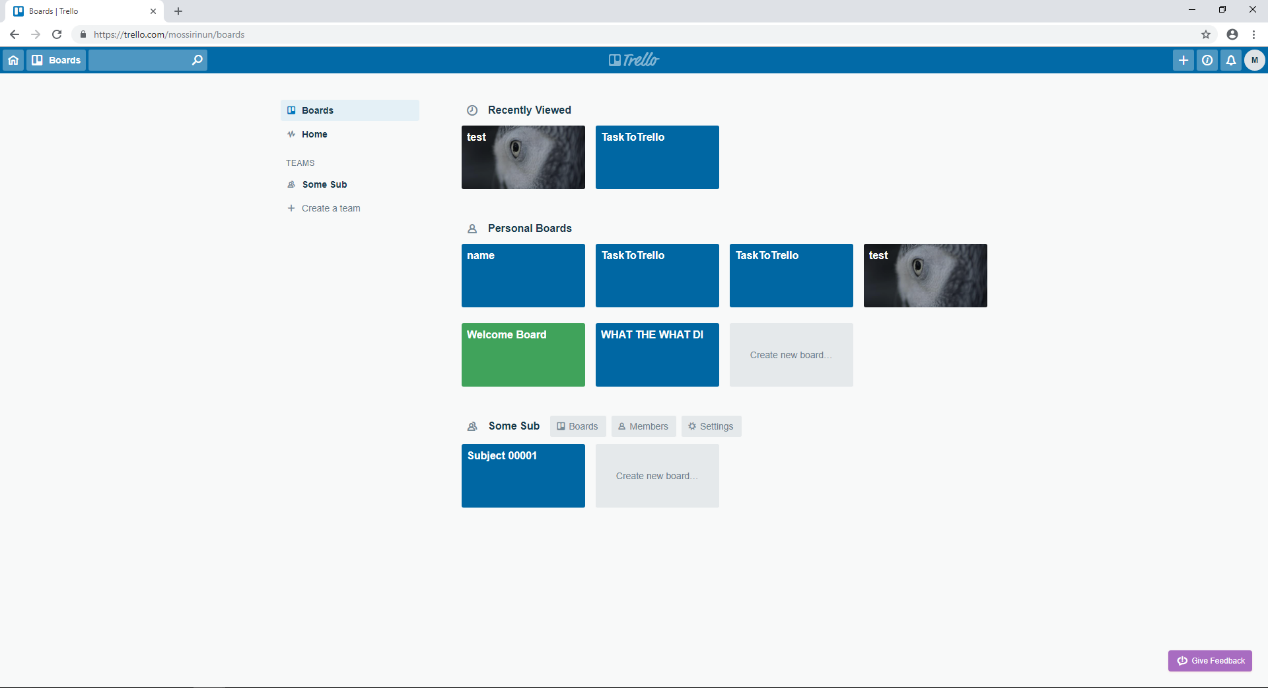
ภาพที่ 23 หน้าจอแสดงผลการเรียกใช้งานผ่าน FileServerFunction

* 1. การทำ Azure Active Directory ให้กับ FileServerFunction
  2. การทำ Azure Content Delivery Network ให้กับ FileServerFunction
  3. ทดสอบการใช้งานหลังจากเซ็ต AD และ CDN
  4. ในการทำฟังก์ชัน task to trello

Trello คือ Collaborative Software เป็นเว็บแอพพลิเคชันที่มีลักษณะเพื่อการทำงานร่วมกันโดยที่ใน Trello นี้เปรียบเสมือนบอร์ดกระดานที่มีรายการงานให้สามารถทำร่วมกัน



ภาพที่ 24 หนจอ board TaskToTrello

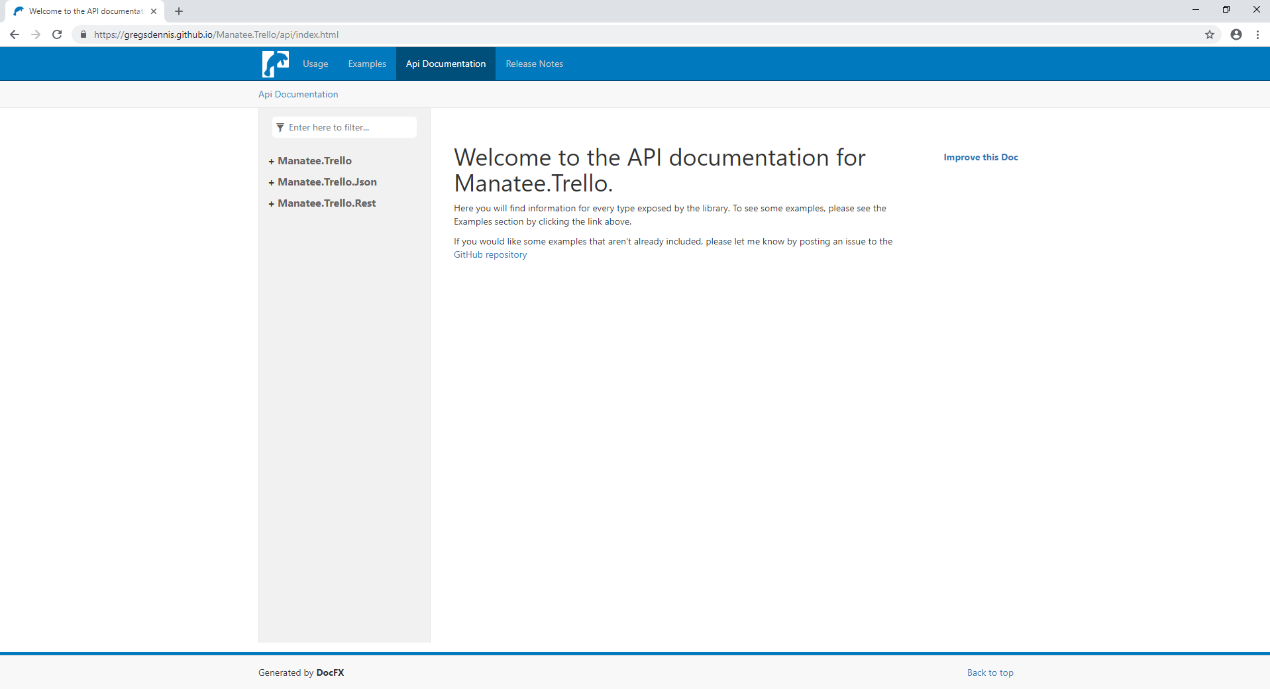


ภาพที่ 25 trello หน้าจอหลัก

จะเห็นว่าในหน้าจอหลักของ trello รายการกระดานอยู่หลากหลายได้แก่ test, TaskToTrello, name, Welcome Board, WHAT THE WHAT DI ทั้งหมดที่กล่าวมาคือรายการกระดานภายใน trello ที่ได้ทำการสร้างไว้ และภายในกระดานนั้นจะมีการ์ดอยู่หลากหลายการ์ดแล้วแต่จำนวนที่สร้าง  
ซึ่งทางผู้จัดทำจะทำการสร้างฟังก์ชันในการรับข้อมูลจากผู้ใช้งานผ่านหน้าฟอร์มแล้วนำข้อมูลที่ได้นั้นไปเพิ่มการ์ดเข้าไปภายใน บอร์ดของ trello

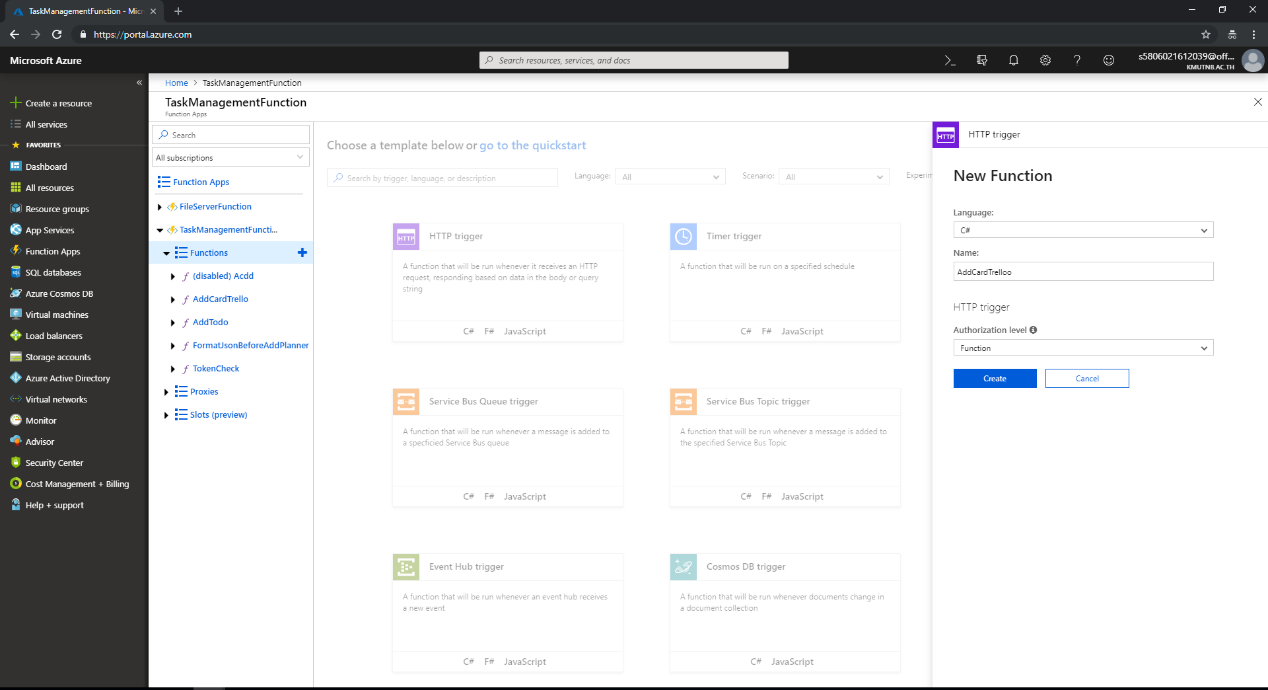
ในการทำฟังก์ชันเพื่อสามารถเขียนคำสั่งให้ใช้งาน Trello ทางผู้จัดทำเลือกใช้งาน API ที่ชื่อ Manatee.Trello โดยจะมีขั้นตอนต่อไปนี้

* ศึกษาการใช้งาน Manatee.Trello



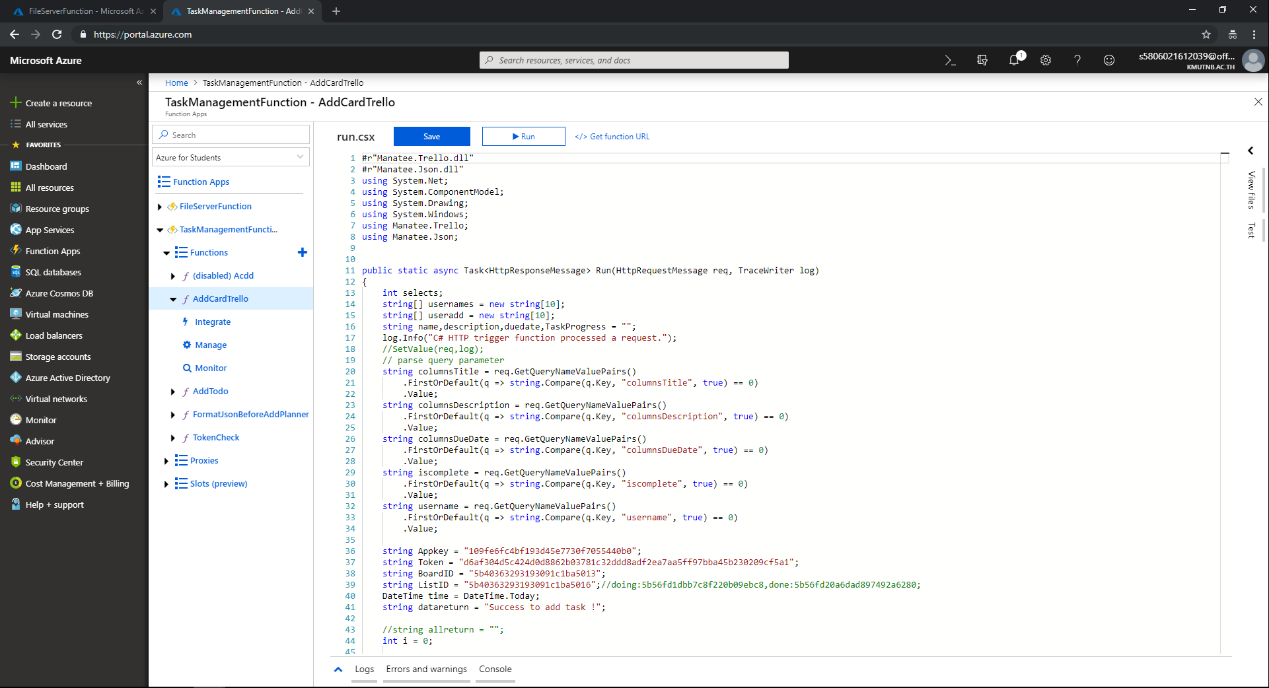
ภาพที่ 26 หนตง manatee.trello document

* หลังศึกษาเร็จแล้วเริ่มทำฟังก์ชันก็ทราบว่าการจะทำให้ใช้งาน trello ได้นั้นจำเป็นต้องมี API Key และ Token ซึ่งเอาไว้ใช้สำหรับการเข้าถึง
* เริ่มการสร้างฟังก์ชันส่วนของการใช้งานกับ trello อันดับแรกเข้ามาที่ Function Apps ที่ชื่อ TaskManagementFunction ที่สร้างไว้ในหัวข้อ 1.2
* ไปที่ Functions เลือก New function
* สร้างฟังก์ชันแบบ HTTP trigger เลือกเป็นภาษา C# แล้วตั้งชื่อฟังก์ชันว่า AddCardTrello



ภาพที่ 27 create new function

* หลังจากสร้างเสร็จแล้วฟังก์ชันจะไปแสดงอยู่ในหัวข้อ “Functions” กดเข้าไปที่ฟังก์ชันที่สร้างหลังจากนั้นทำการเขียนคำสั่งให้สามารถรับข้อมูลมาจากผู้ใช้งานและส่งค่าข้อมูลเหล่านั้นไปสร้างภายใน trello



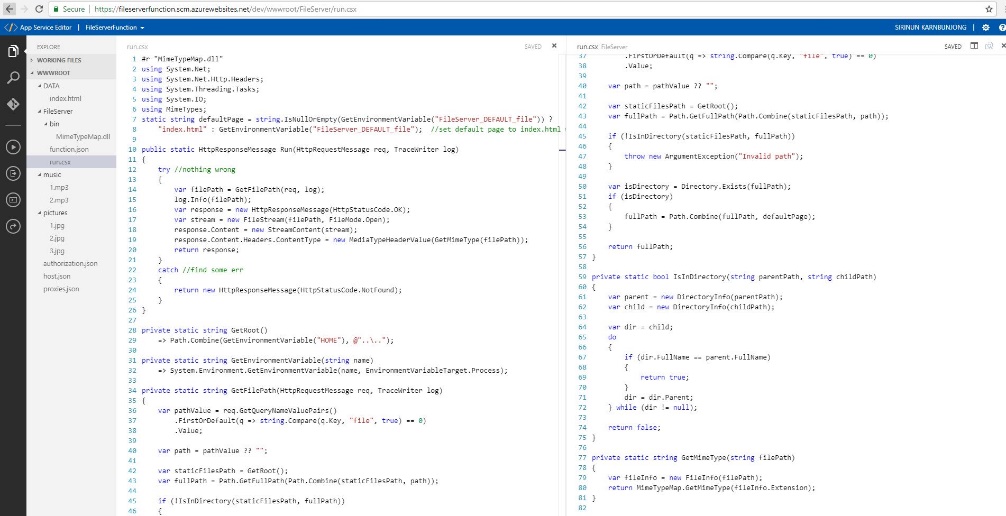
ภาพที่ 28 AddCardTrello function code preview

* รายละเอียดการใช้งาน api เช่น key token หามายังไง
* เริ่มเขียนคำสั่งในการเพิ่มtask
  1. ในการทำฟังก์ชัน Task to Microsoft Planner
* ทดลองใช้งาน
* ค้าคว้าหา api ที่จะนำมาใช้ในการทำงานด้วย
* รายละเอียดการใช้งาน api ทำยังไงถึงจะใช้ได้
  1. ในการทำฟังก์ชัน task to Microsoft to-do
* ทดลองใช้งาน
* ค้าคว้าหา api ที่จะนำมาใช้ในการทำงานด้วย
* รายละเอียดการใช้งาน api ทำยังไงถึงจะใช้ได้

#### **ศึกษาการใช้งาน** Microsoft Azure Function App สำหรับการทำ File Server Function

* 1. รวบรวมข้อมูล ค้นคว้าในการสร้าง **File Server Function**

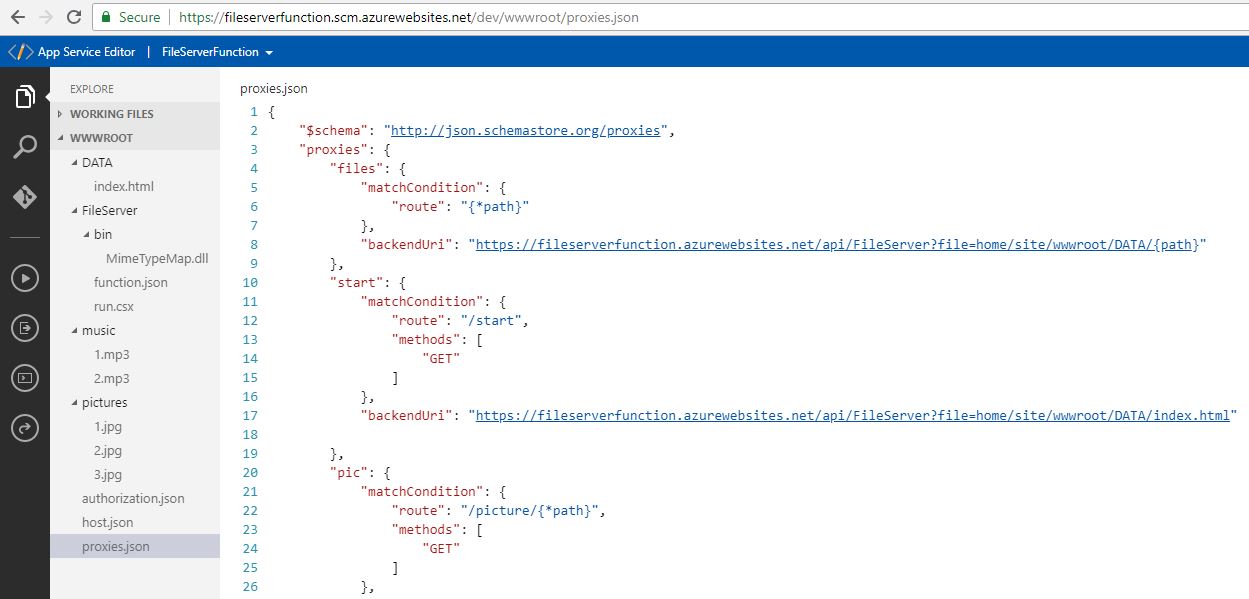
**เริ่มจากการสร้างฟังก์ชันขึ้นมาแล้วค้นคว้าหาวิธีการทำฟังก์ชันนี้ให้สามารถเรียกใช้งานไฟล์ที่ถูกเก็บไว้ภายในแสดงออกมาได้**

**จนได้มาพบกับงานของบุคคลหนึ่ง ที่ได้ทำโครงสร้างของ** File Server Function **เอาไว้เรียบร้อยแล้วโดยจะนำโค้ดคำสั่งมาประยุกต์ ให้เข้ากับการใช้งานที่ดำเนินการ**

ภาพที่ 29 แสดงตัวอย่างคำสั่งในหน้า File Server Function

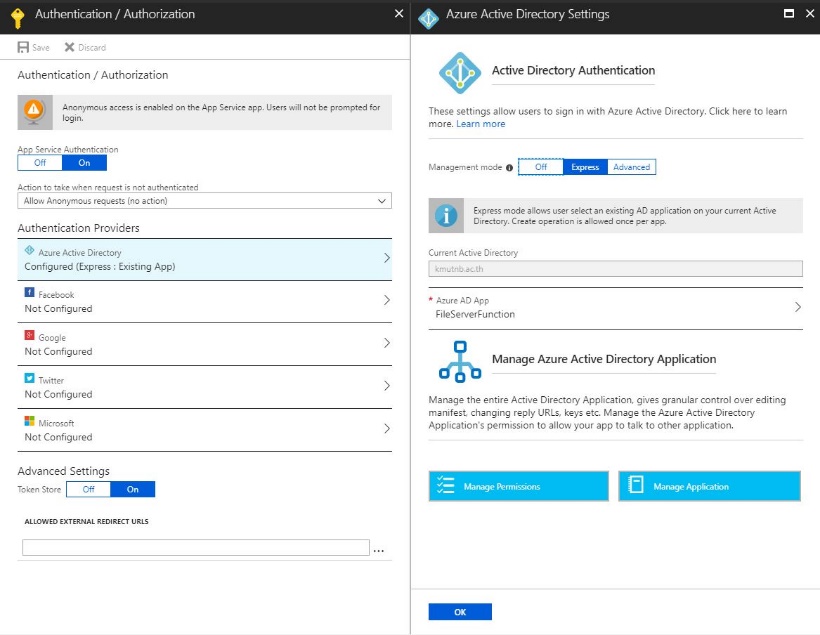
* 1. ศึกษาการทำงานของ **Proxies** ที่อยู่ภายใน **Microsoft Azure Function App**

**สร้างฟังก์ชัน** Proxies **เพื่อเป็นตัวเรียกใช้งานไฟล์ที่ถูกเก็บไว้ภายใน** File Server-Function **ให้ออกมาแสดงที่หน้าเว้บไซต์ โดยมีวิธีการดังนี้**

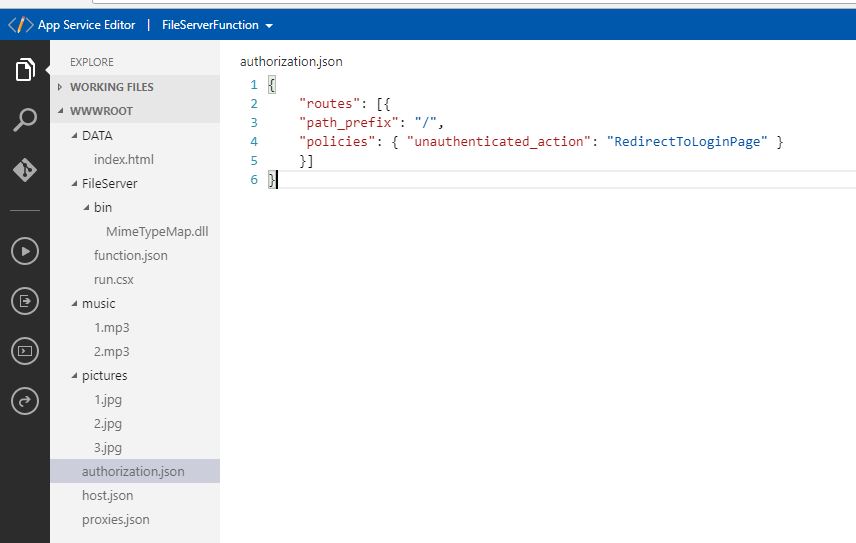
* + - * **กำหนด** path **ที่ได้ผ่านเข้ามาให้ไปเรียกใช้งานไฟล์ที่อยู่ใน** File Server
      * **กำหนด** backendURL **ให้จับคู่กับ** path **หากตรงกันก็จะทำงาน**

ภาพที่ 30 แสดงตัวอย่างในหน้า Proxies

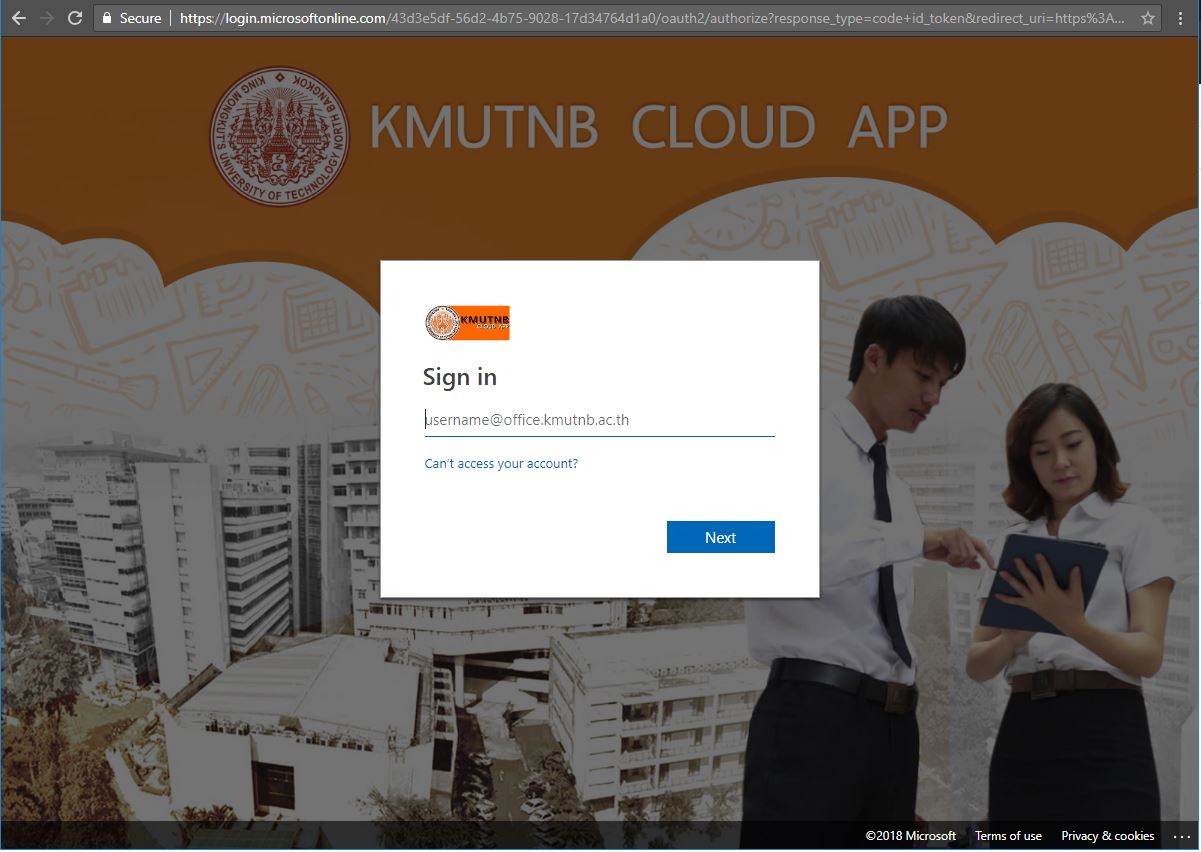
* 1. ใช้ **Azure Active Directory** ให้ทำงานร่วมกับ **File Server Function App**
     1. **ศึกษาวิธีการใช้งานของ** Azure Active Directory **เพื่อนำมาประยุกต์ใช้งานร่วมกับ** File Server Function **โดยจะมีขั้นตอนดังนี้**
        + **เปิดใช้งาน** Authentication / Authorization **ใน** File Server Function
        + **กำหนดกลุ่มของผู้ใช้งานที่อนุญาตให้เข้าถึงข้อมูล**



ภาพที่ 31 แสดงตัวอย่างการเปิดใช้งาน Authentication / Authorization

* + - * **สร้างไฟล์ใน** Fil Server Function **เพื่อทำให้เป็นการกำหนดการ** Authorization **ถ้าหากเข้ามา** path**จะให้ทำการ** redirect **ไปยังหน้า** login

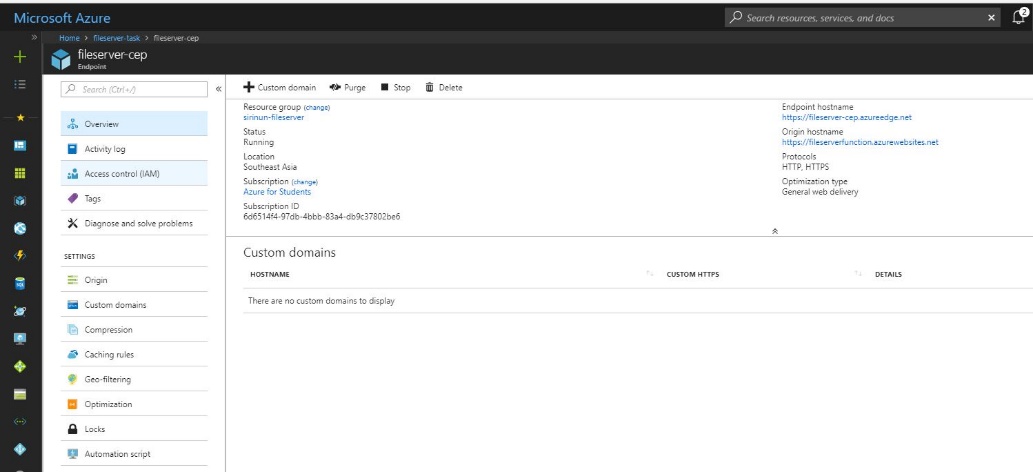
ภาพที่ 32 แสดงตัวอย่างหน้าไฟล์ Authorization

* 1. ใช้**Azure Content Delivery Network** ให้ทำงานร่วมกับ **File Server Function App**

ภาพที่ 33 แสดงตัวอย่างหน้า Login เข้าใช้งานไฟล์

* + 1. **ศึกษาวิธีการใช้งานของ** Azure Active Directory **เพื่อนำมาประยุกต์ใช้งานร่วมกับ**File Server Function **โดยจะมีวิธีการดังนี้**
       - **สร้าง** Azure Content Delivery Network **เมื่อสร้างเสร็จจะได้หน้า** Endpoint

ภาพที่ 34 แสดงตัวอย่างหน้าที่ใช้สร้าง CDN

* + - * **ในหน้า** CDN-Endpoint **จะมีลิงค์** Endpoint **แสดงให้เรานำมาใช้งานได้**

ภาพที่ 35 แสดงตัวอย่างหน้า Endpoint



ภาพที่ 36 แสดงตัวอย่างหน้าการใช้งานจากลิงค์(จะเห็นว่าลิงค์นั้นเป็น .azureedge.net)

#### ศึกษาการใช้งาน Microsoft Azure Function App สำหรับการทำ Trello Function

* 1. ศึกษาการรับข้อมูลจากผู้ใช้งาน

**เมื่อผู้ใช้งานได้ส่งผ่านเข้ามาทำงานภายในฟังก์ชัน จะต้องทำการเก็บค่าข้อมูลแหล่านั้นไว้ในตัวแปร และนำไปประมวลผลในการเพิ่มรายการงานเข้าไปใน** Trello , Microsoft To-Do , Microsoft Planner **ตามลำดับ**

ภาพที่ 37 แสดงตัวอย่างหน้าฟอร์มในการรับข้อมูล

* 1. ศึกษาการส่งข้อมูลกลับคืน

**เมื่อทำการประมวลผลข้อมูลที่ได้ตามต้องการแล้ว จะต้องทำการ** Response **ข้อมูลที่ต้องการแสดงกลับไปยังหน้าฟอร์มเพื่อให้ผู้ใช้งานได้ทราบว่าระบบทำงานเสร็จสมบูรณ์หรือไม่**

* 1. ศึกษาค้นคว้าวิธีการ **Add card into Trello**

**ภายในฟังก์ชันการทำงานจะใช้ภาษา** C# **ในการพัฒนาระบบจึงได้ค้นคว้าหาวิธีการ** Addcard **เข้าไปภายใน** Trello **จนไปพบกับ** Manatee.Trello **โดยเป็น** API **ตัวช่วยในการทำงานในครั้งนี้**

**โดยในการทำนั้นจะต้องมีข้อมูลของ Trello ดังนี้**

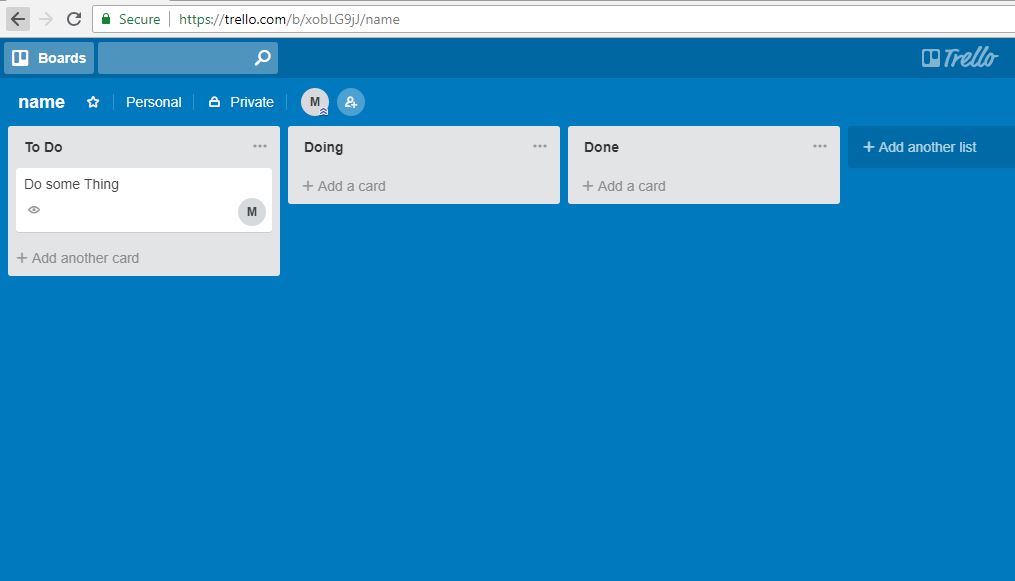
* + - * + Token
        + AppKey
        + BoardID **ของ** Trello
        + List ID **ของ** Trello

**หลังจากทราบข้อมูลข้างต้นแล้ว ก็จะเริ่มทำการเพิ่มข้อมูลโดยเริ่มจาก**

**เพิ่มการ์ดเข้าไปภายใน** Trello **โดยในการ์ดนั้นจะมีข้อมูลที่ถูกเพิ่มไปพร้อมกันคือ** name, description, duedate, iscomplete **ตามลำดับ**

**ในระหว่างการเพิ่มการ์ดนั้นจะทำการเพิ่มสมาชิกที่เกี่ยวข้องเข้าไปภายในการ์ดที่ถูกสร้างขึ้น**

ภาพที่ 38 แสดงตัวอย่างของ Trello

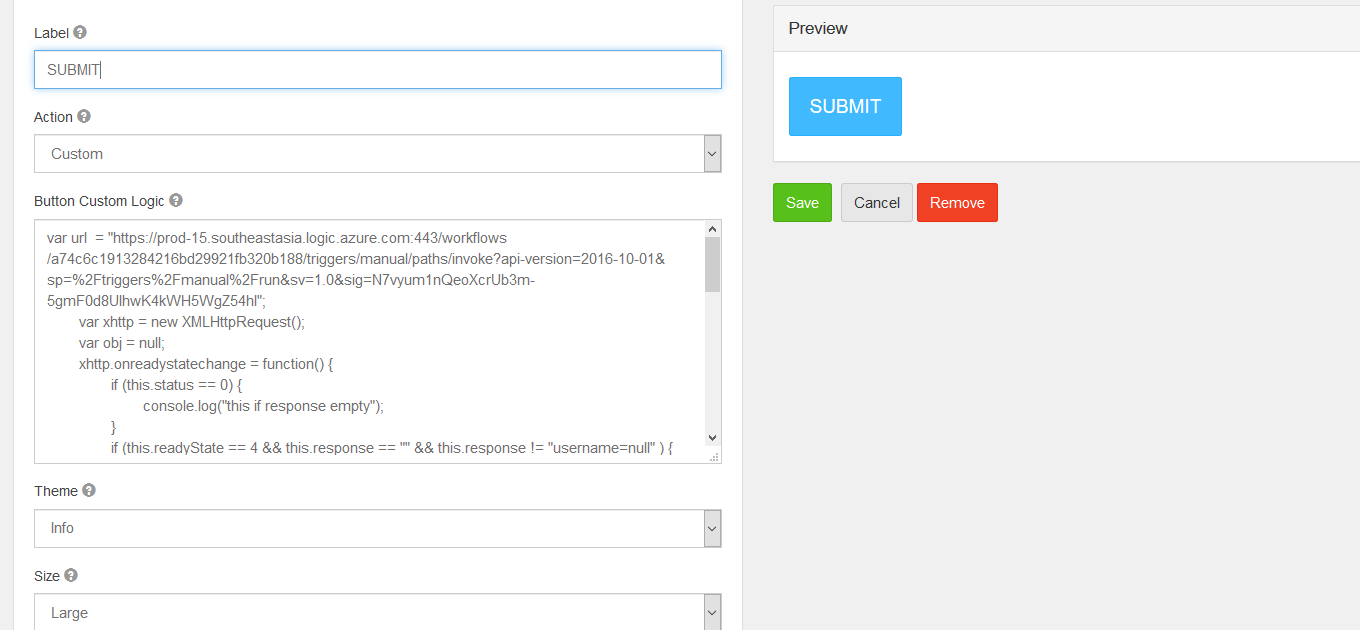
****

#### ศึกษาการทำหน้าฟอร์ม โดยใช้ Form.IO ให้รันผ่านไฟล์ HTML

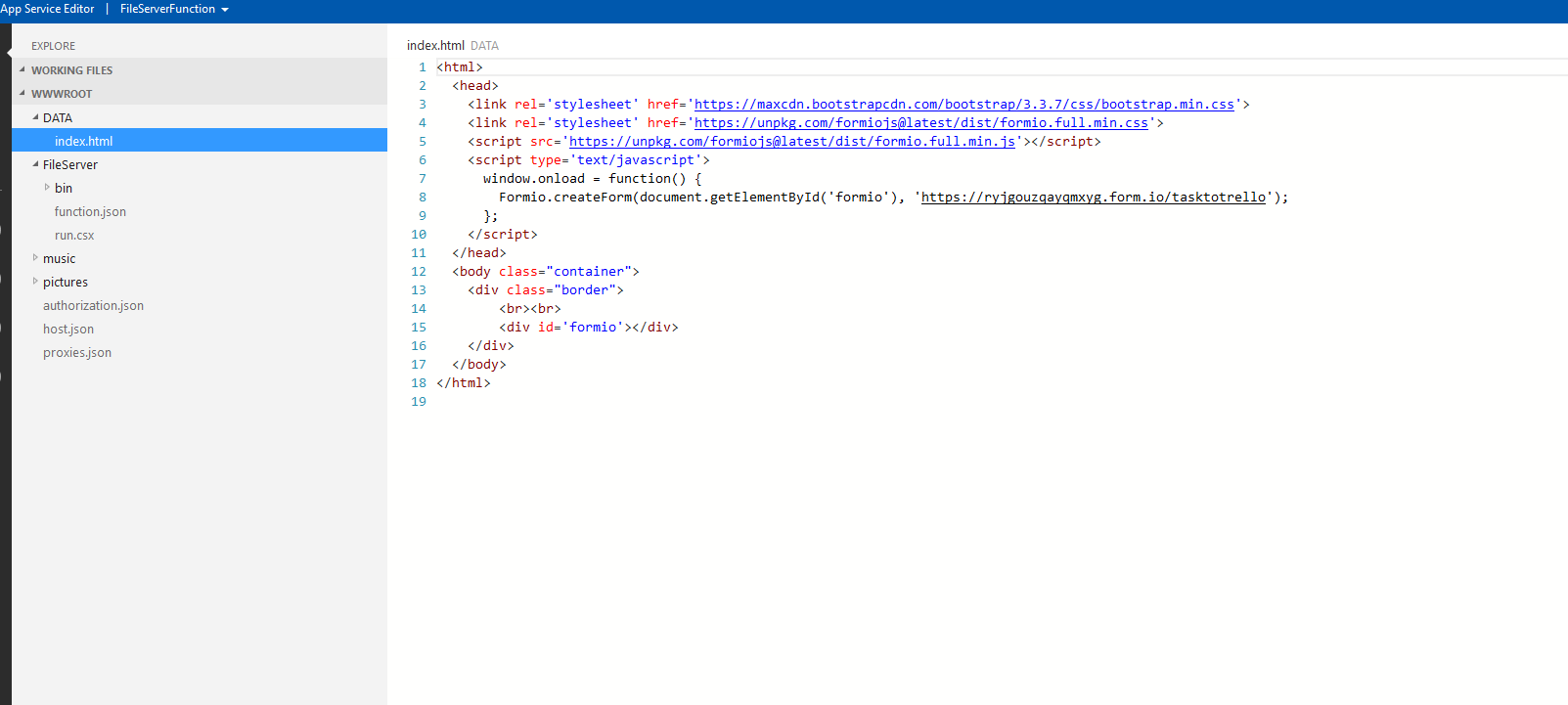
ทำการกำหนดหน้าตาของแบบฟอร์มที่ต้องการให้ผู้ใช้งานกรอกข้อมูลในระบบของเรา แล้วนำข้อมูลเหล่านั้นส่งเข้าไปทำงานภายในฟังก์ชันตามที่เราต้องการ โดยมีวิธีการดังนี้

* 1. เมื่อเข้ามาอยุ่ในหน้าเว็บของ Form.IO แล้วก็เริ่มทำการออกแบบฟอร์มตามที่ต้องการได้ โดยการลาก Components มาวางใส่ได้เลย เช่น Textbox, select box เป็นต้น

ภาพที่ 39 แสดงตัวอย่างของ components ใน Form.IO

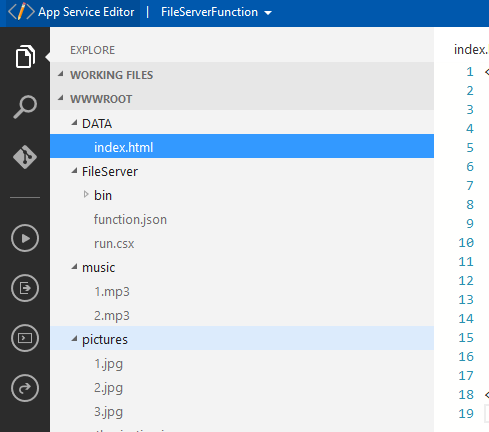
* 1. หลังจากทำการวาง Components ได้ตามที่ต้องการแล้ว ก็ทำการเซ็ตค่าคำสั่งที่ปุ่ม Submit โดยคำสั่งจะเป็น เมื่อทำการกดปุ่มแล้วให้เกิดการทำงานโดยข้อมูลที่กรอกลงไปนั้นจะถูกส่งเข้าไปทำงานใน LogicApp แล้วให้ส่งข้อมูลไปยังฟังก์ชันต่อไป

ภาพที่ 40 แสดงตัวอย่างของคำสั่งปุ่ม submit ใน Form.io

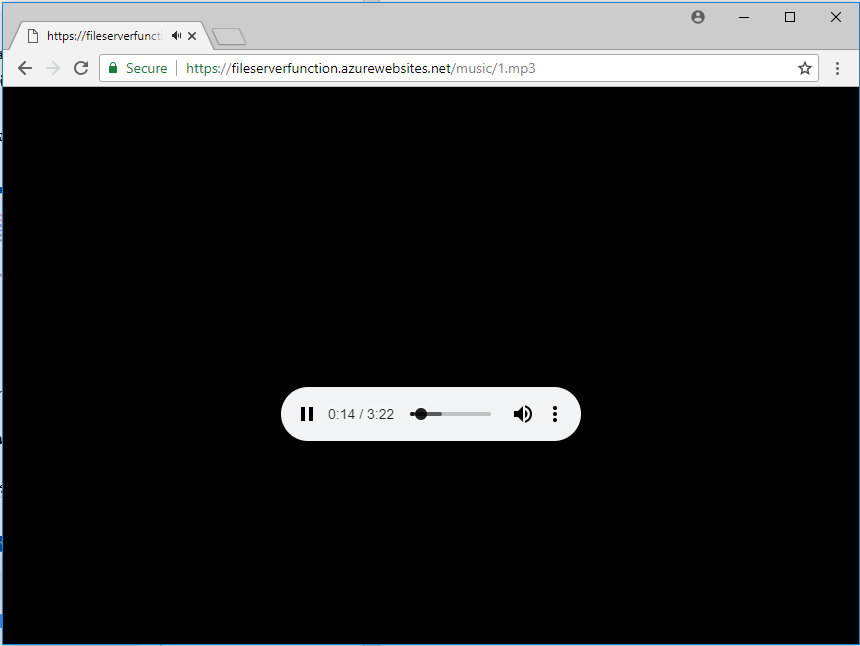
* 1. หลังจากสร้างหน้าฟอร์มเสร้จสมบูรณ์แล้วให้ทำการสร้างไฟล์ HTML ขึ้นมาให้ทำการ Rendering Form.IO ได้โดยศึกษาจาก FormIO.JS ซึ่งเป็นโค้ดคำสั่งในการเรียกฟอร์ม  
     จากลิ้ง URL ของ Form.IO หรือจะเรียกจาก JSON ที่ได้จาก Form.IO เหมือนกันแต่ในที่นี้จะเลือกใช้แบบ URL

ภาพที่ 41 แสดงตัวอย่างการเรียกหน้าฟอร์มผ่านไฟล์ HTML

#### ศึกษาวิธีการนำฟอร์มที่ได้ทำการรันในไฟล์ HTML นำไปเก็บไว้ใน File Server Function

เปิดเข้าไปภายในฟังก์ชั้นเลือกเปิดแบบโหมด Editor แล้วทำการอัพโหลดไฟล์เข้าไปเก็บเอาไว้ภายในตามต้องการ

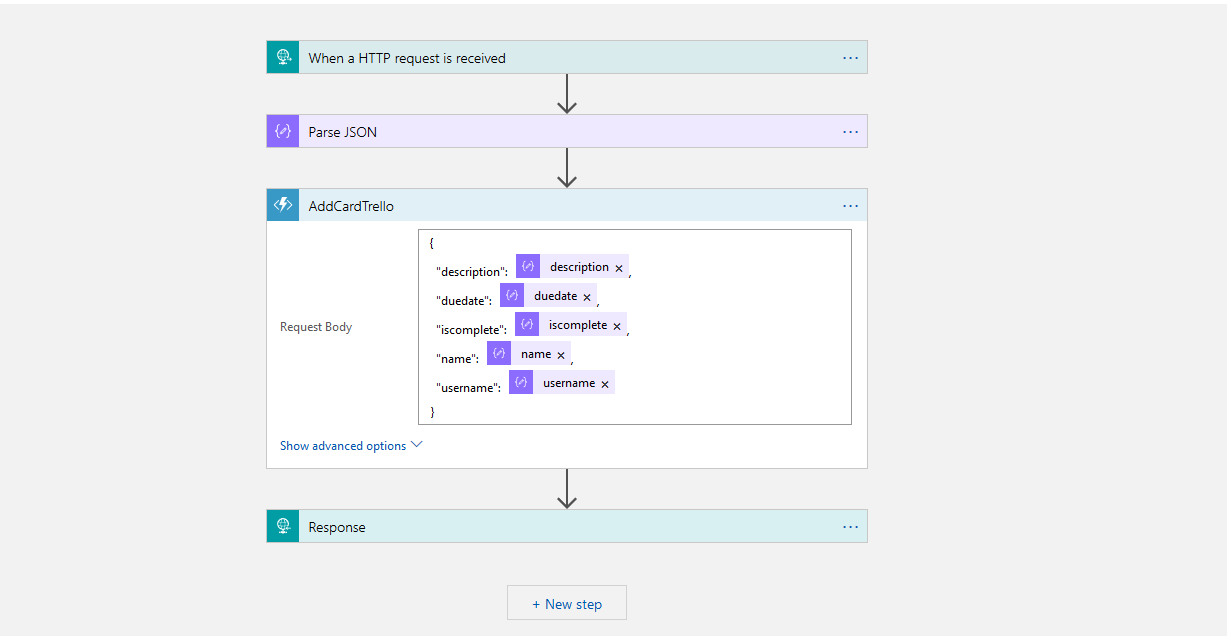
ภาพที่ 42 แสดงตัวอย่างไฟล์ที่อัพโหลดลง File Server Function

* 1. เมื่อนำไปใส่แล้วก็ทดลองเรียกใช้งานไฟล์ที่ถูกเก็บ ถ้ามีการแสดงผลก็สำเร็จ ตัวอย่างจากที่ได้อัพโหลดไฟล์เพลงและภาพลงไปแล้วก็ลองทำการเรียกมาแสดงผลสักไฟล์นึง โดยจะทำการเรียกใช้งานไฟล์**เพลง**ออกมาแสดง

ภาพที่ 43 แสดงตัวอย่างการเรียกไฟล์เพลง

#### ศึกษาวิธีการทำ Azure logic app เพื่อจัดการข้อมูลที่ได้มาจากหน้าฟอร์มส่งไปยังฟังก์ชัน

เริ่มทำการสร้าง Logic App ขึ้นมาแล้วเพิ่มขั้นตอนการทำงานเข้าไปได้โดยจะทำการเรียงลำดับการทำงาน Request-ตัวแปรเก็บข้อมูล-ส่งข้อมูลเข้าฟังก์ชัน-Response  
ตามลำดับโดยดูตัวอย่างได้ตามรูป



ภาพที่ 44 แสดงตัวอย่าง Logic App เชื่อมต่อกับ File Server Function

# **บทที่ 4 ผลการดำเนินงาน**

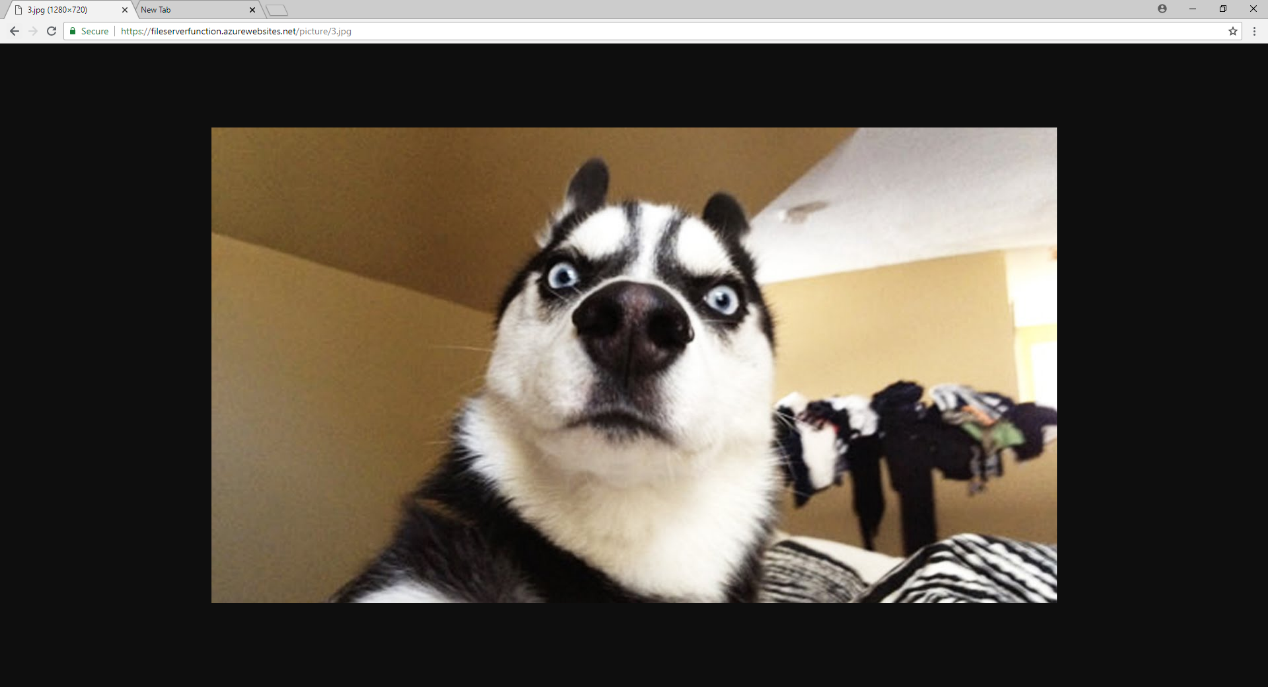
ผลการดำเนินงานในการพัฒนาเว็บแอปพลิเคชัน File Server และ Task Management

ได้พัฒนาตามขอบเขตเป็นที่เรียบร้อย และเป็นไปตามวัตถุประสงค์ของการพัฒนา โดยมีส่วนการทำงานดังนี้

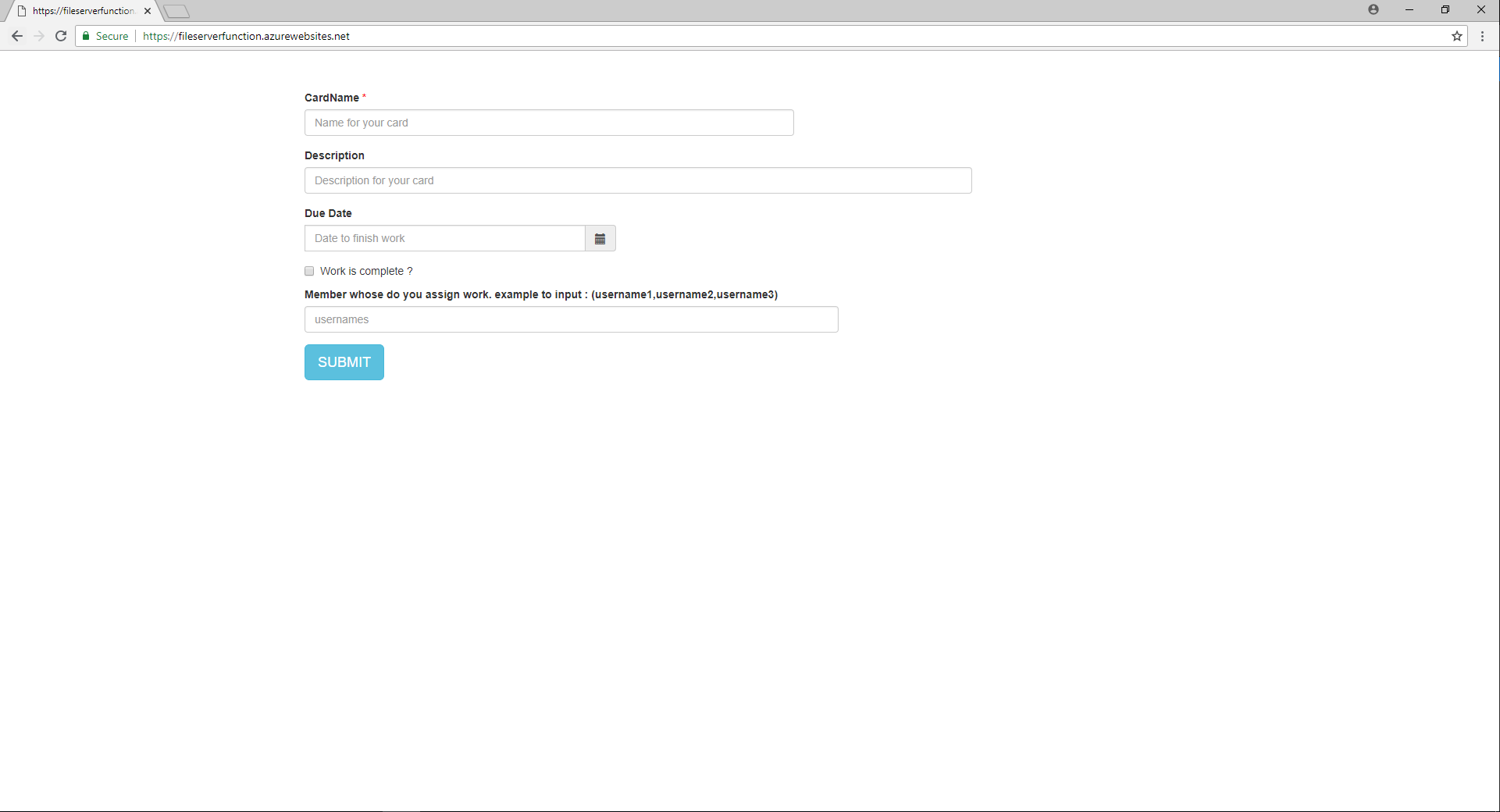
## สามารถเรียกใช้งานไฟล์ผ่าน File Server Function ได้

* 1. ******เรียกใช้งานไฟล์เพลงที่อยู่ใน File Server Function ออกมาแสดงผล สามารถทำงานได้ตามปกติ**

ภาพที่ 45 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง

* 1. **เรียกใช้งานไฟล์รูปภาพที่อยู่ใน File Server Function ออกมาแสดงผล สามารถทำงานได้ตามปกติ**

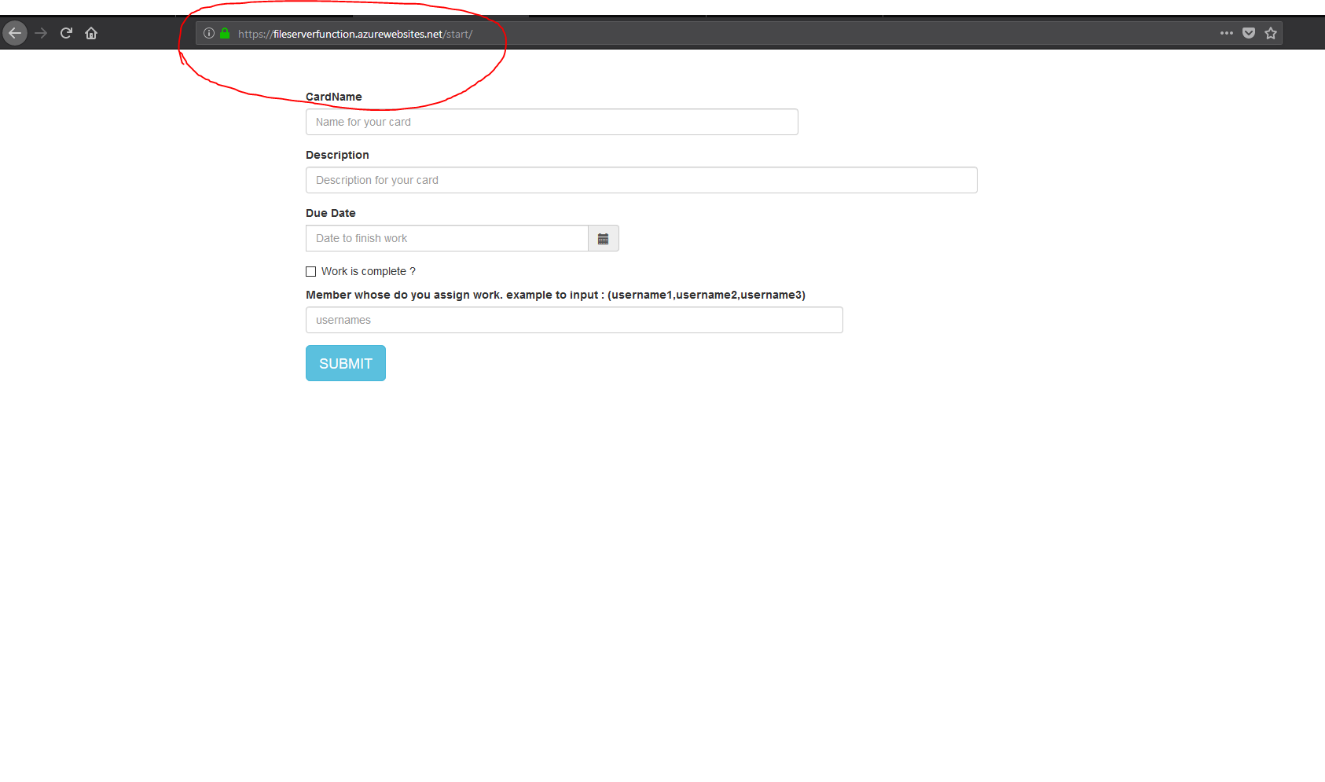
ภาพที่ 46 แสดงตัวอย่างการเรียกใช้งานไฟล์รูปภาพ

* 1. **เรียกใช้งานไฟล์ index.html ที่อยู่ใน File Server Function ออกมาแสดงผล สามารถทำงานได้ตามปกติ**

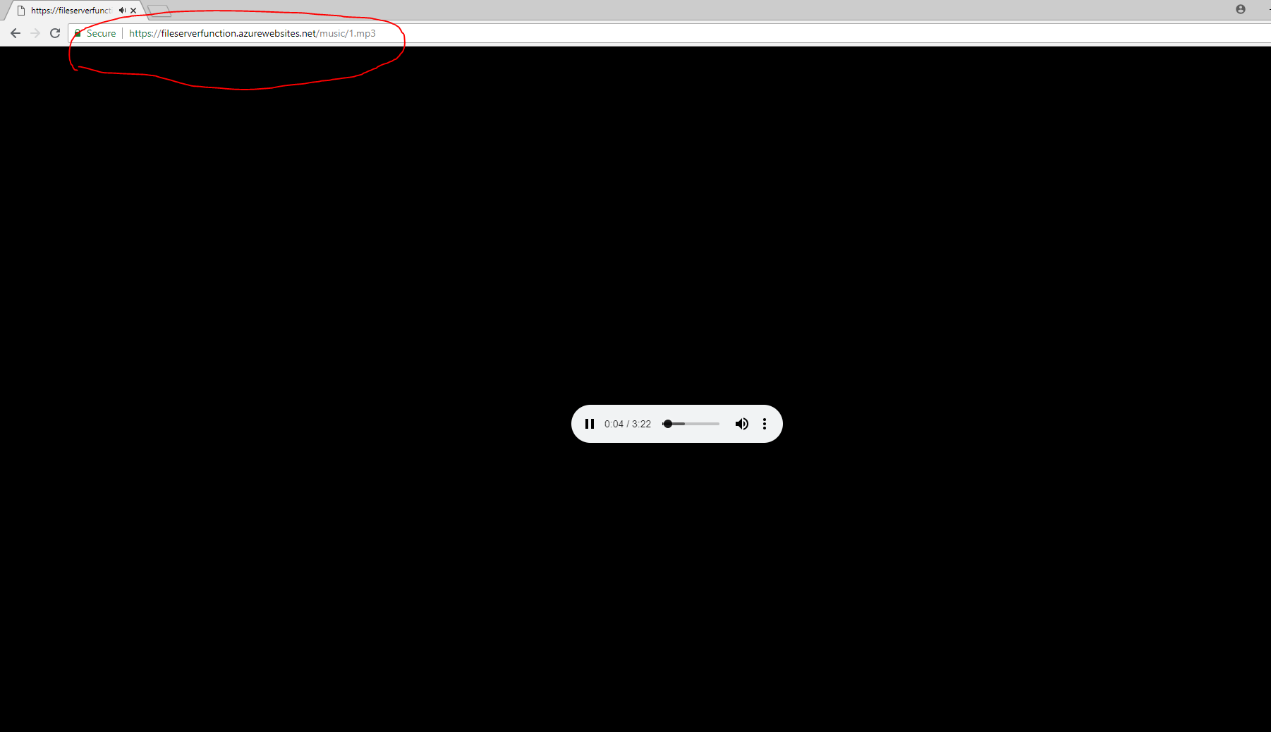
ภาพที่ 47 แสดงตัวอย่างการเรียกใช้งานไฟล์ index.html

## สามารถเรียกใช้งานไฟล์ผ่าน File Server Function โดยเรียกผ่าน CDN ได้

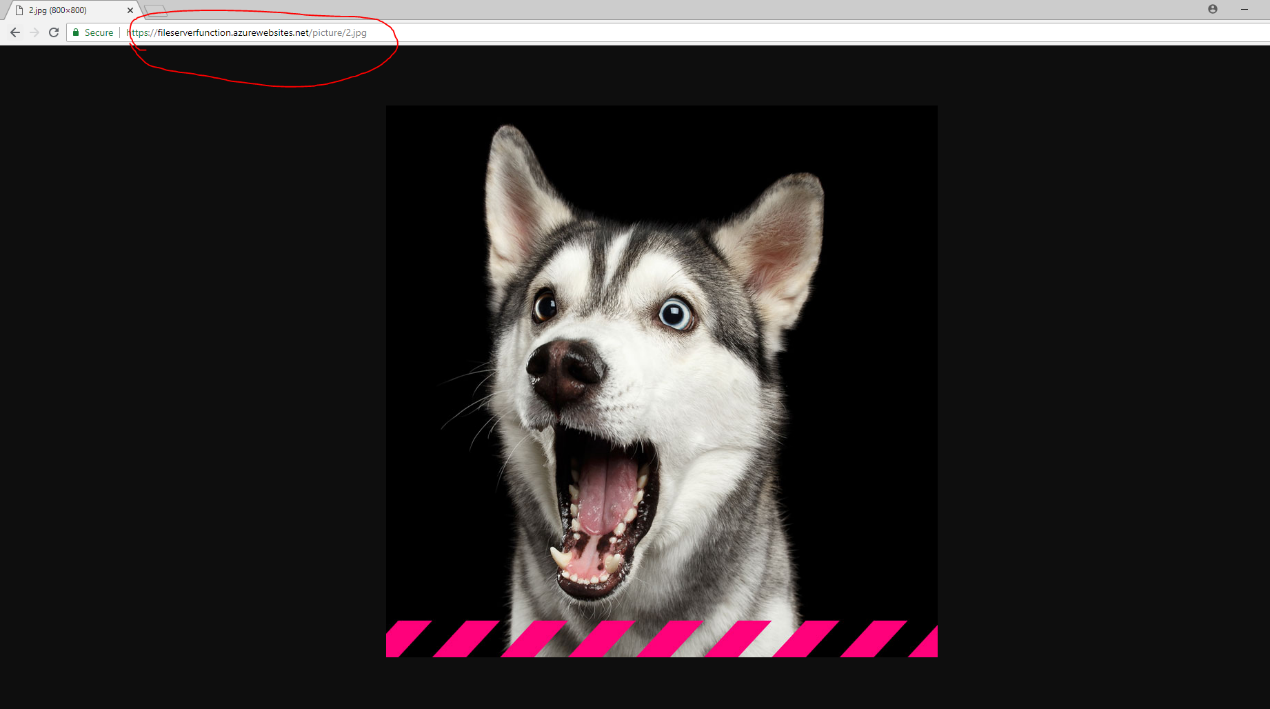
* 1. **เรียกใช้งานไฟล์ index.html ที่อยู่ใน File Server Function ผ่าน CDN ออกมาแสดงผล สามารถทำงานได้ตามปกติ สังเกตจากลิงค์ที่เป็น .azurewebsite.net**



ภาพที่ 48 แสดงตัวอย่างการเรียกใช้งานไฟล์ index.html ผ่าน CDN

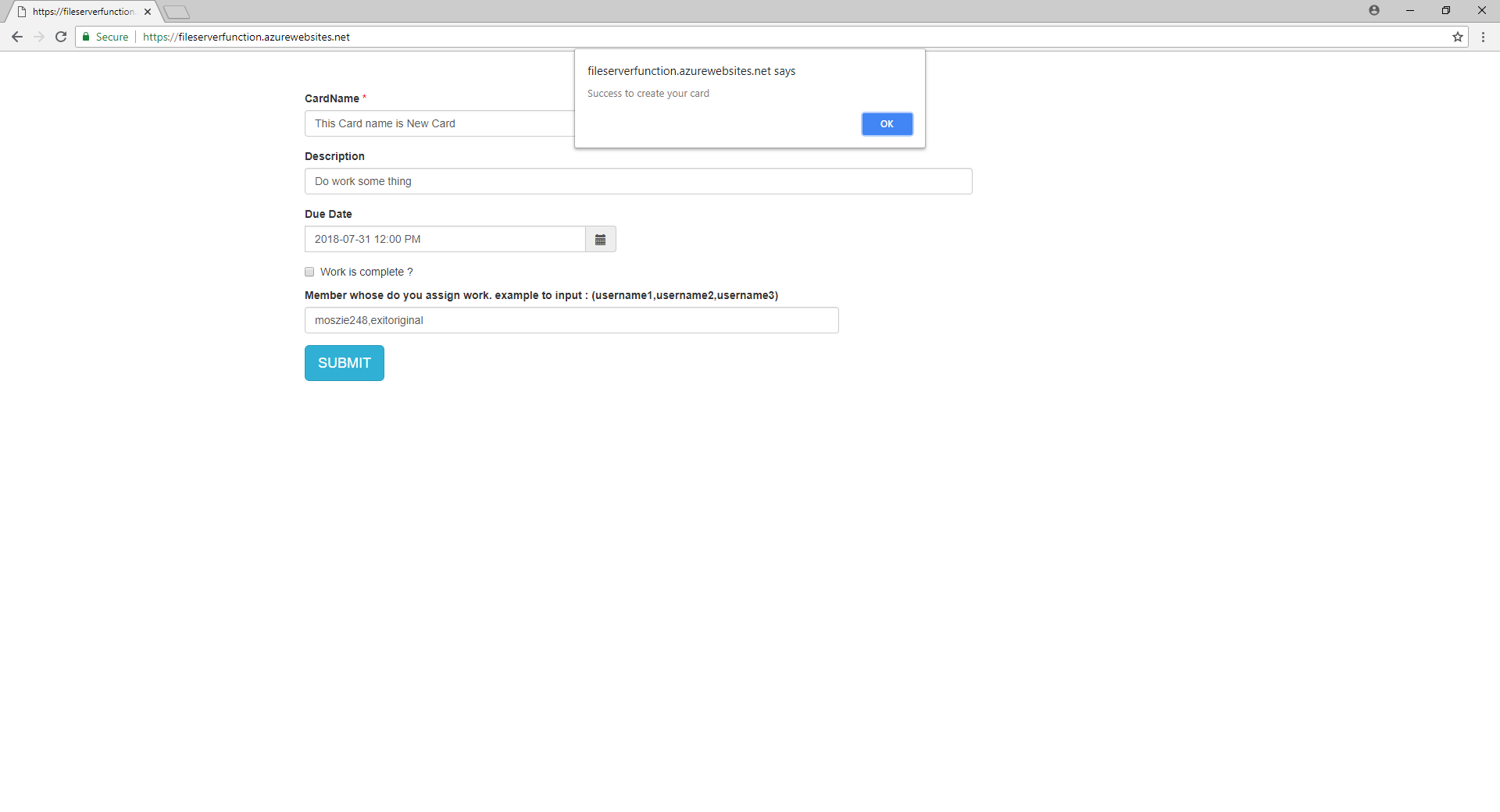
* 1. **เรียกใช้งานไฟล์เพลงที่อยู่ใน File Server Function ผ่าน CDN ออกมาแสดงผล สามารถทำงานได้ตามปกติ สังเกตจากลิงค์ที่เป็น .azurewebsite.net**

ภาพที่ 49 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง ผ่าน CDN

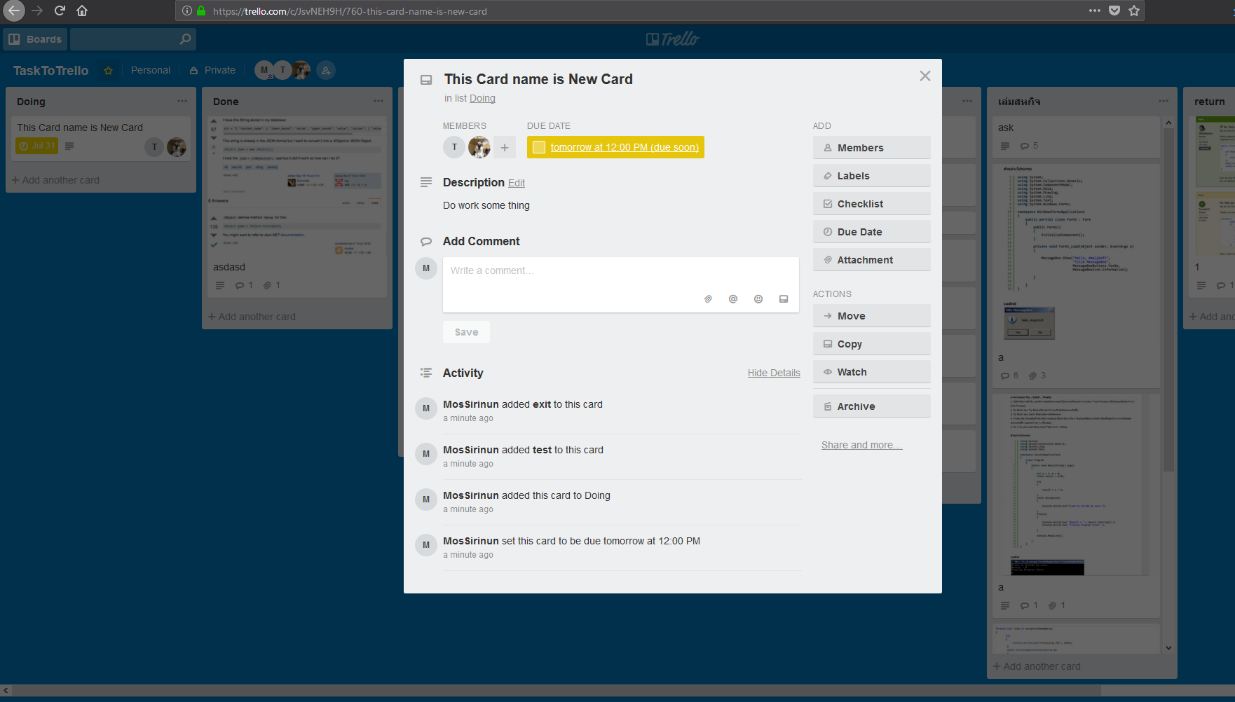
* 1. **เรียกใช้งานไฟล์รูปภาพที่อยู่ใน File Server Function ผ่าน CDN ออกมาแสดงผล สามารถทำงานได้ตามปกติ สังเกตจากลิงค์ที่เป็น .azurewebsite.net**

ภาพที่ 50 แสดงตัวอย่างการเรียกใช้งานไฟล์เพลง ผ่าน CDN

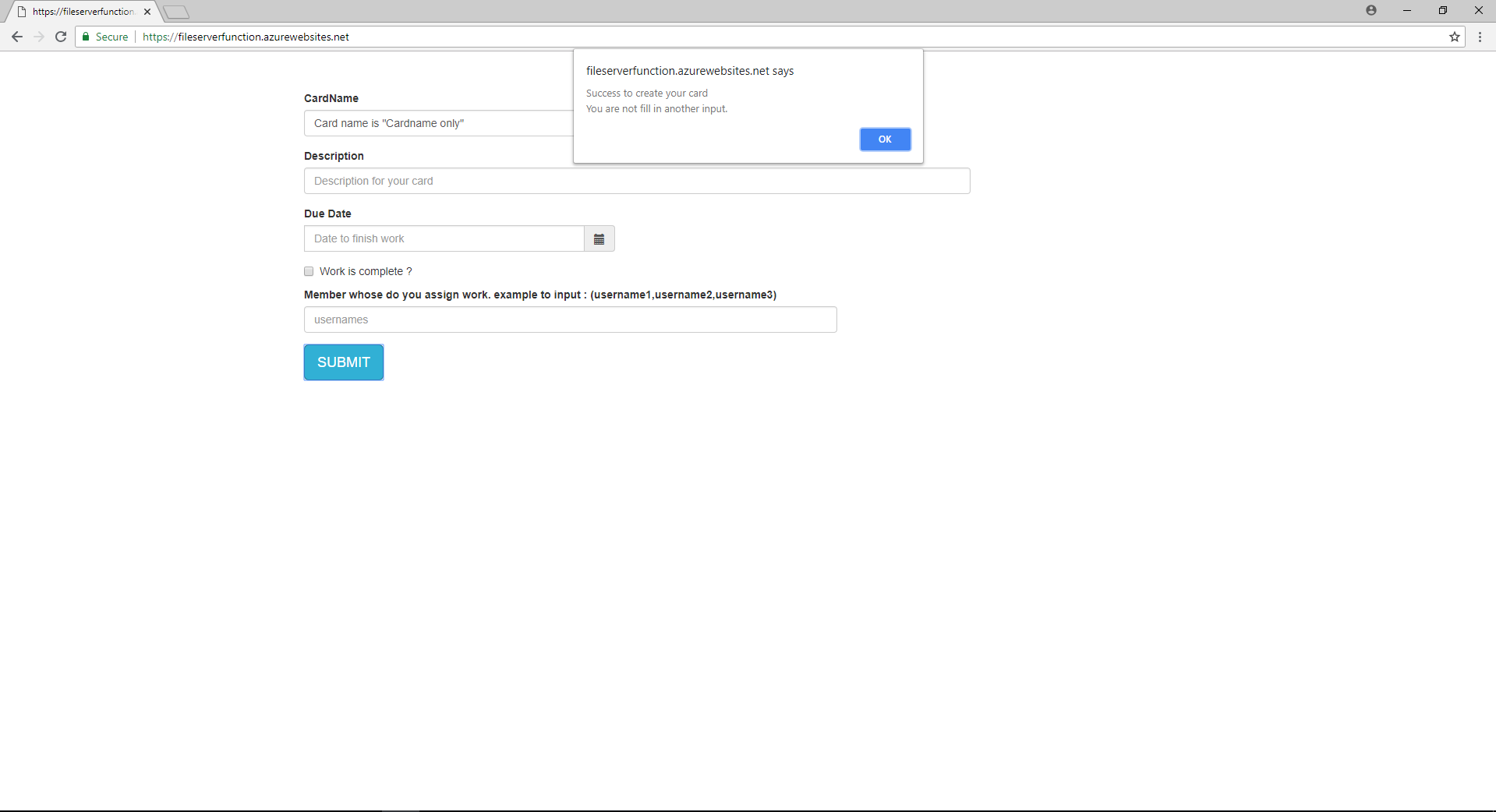
## สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จ

* 1. **สร้าง card เมื่อทำการกรอกข้อมูลในหน้าฟอร์มถูกตามหลักครบทุกค่า  
     เมื่อกดปุ่ม submit จะมี alert แจ้งเตือนว่าสำเร็จ**

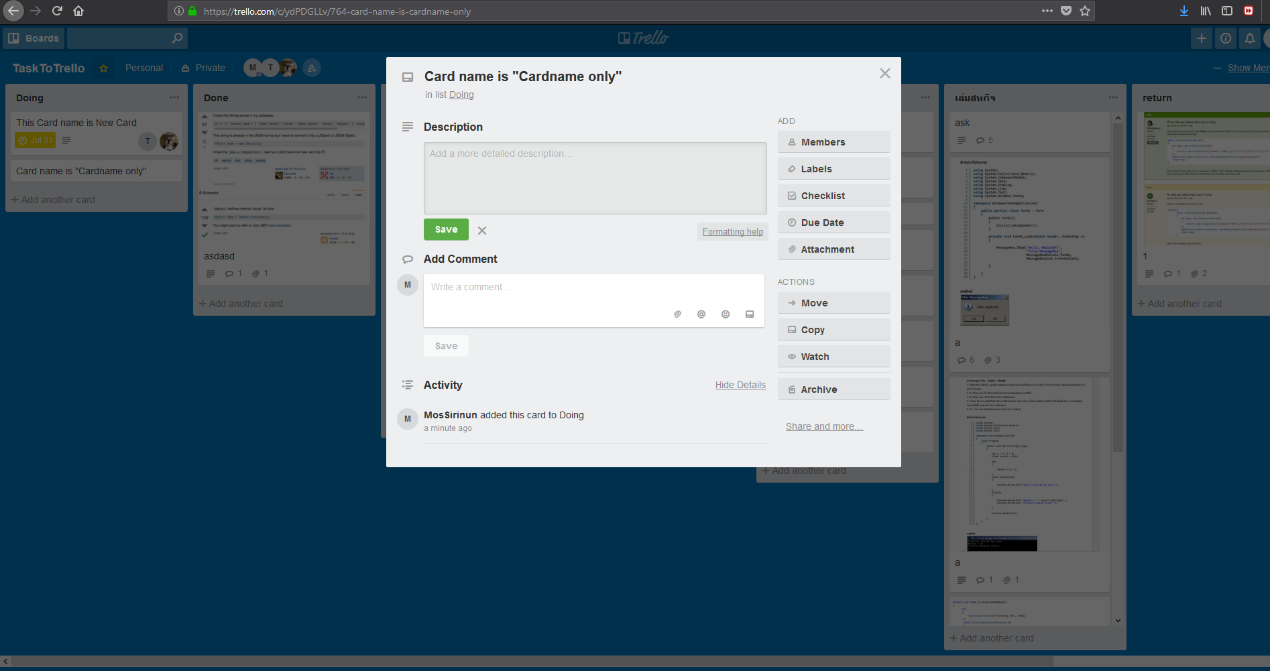
ภาพที่ 51 แสดงตัวอย่างการกรอก input แล้วสำเร็จ

* 1. **เมื่อกดปุ่ม Submit แล้วข้อมูลที่ถูกกรอกเข้าไปจะถูกนำไปสร้าง Card ใน Trello ได้สำเร็จ**

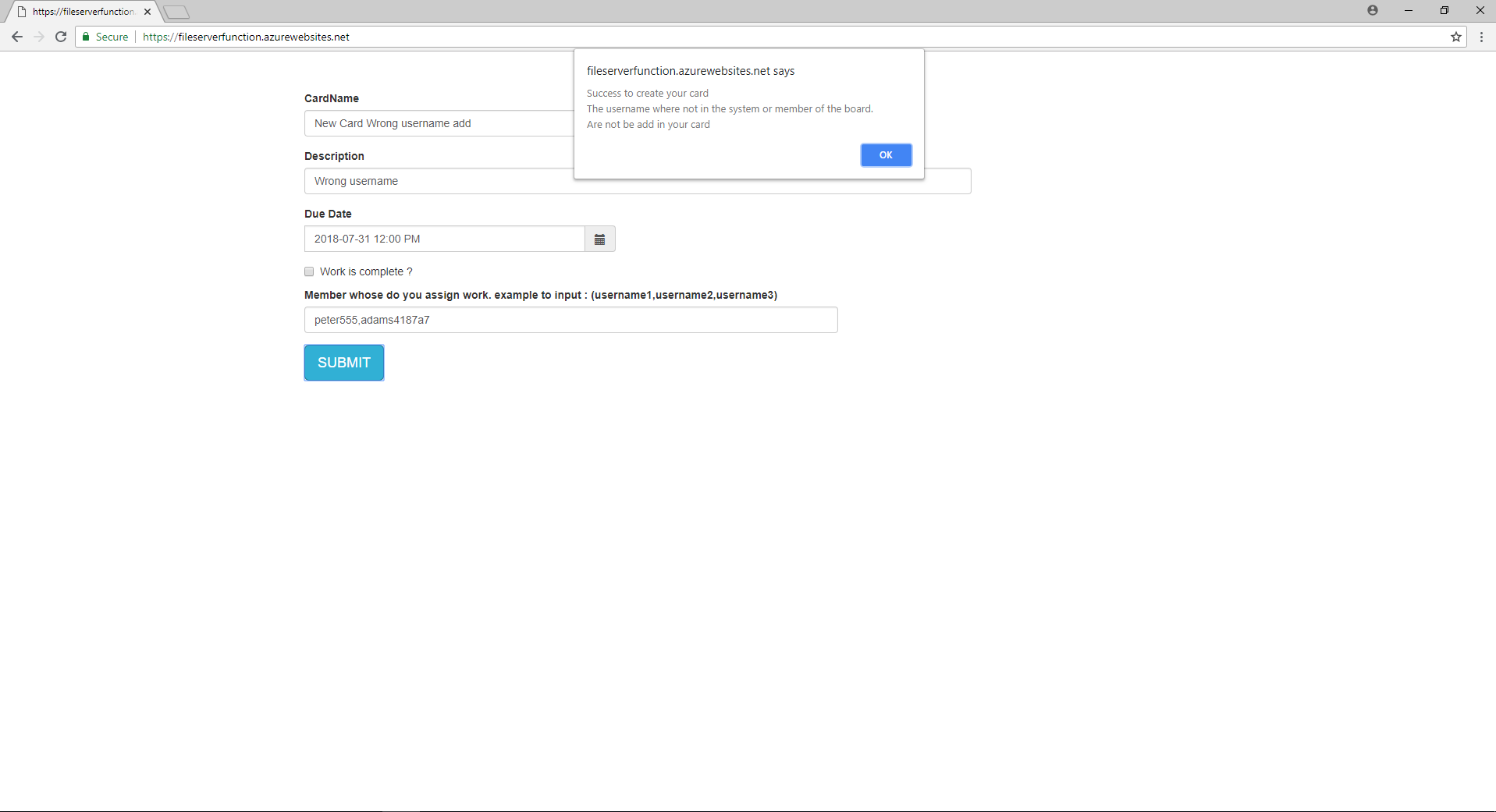
ภาพที่ 52 แสดงตัวอย่าง card ใน Trello ตามข้อมูลที่ได้รับ

1. สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จเมื่อกรอกค่า card name ค่า**เดียวโดยปกติแล้วการสร้าง card นั้นจะมีแค่ชื่อของการ์ดก็สามารถสร้างได้สำเร็จ**
   1. ******สร้าง card เมื่อทำการกรอกข้อมูลในหน้าฟอร์มเพียงแค่ค่า card name  
      เมื่อกดปุ่ม submit จะมี alert แจ้งเตือนว่าสำเร็จ**

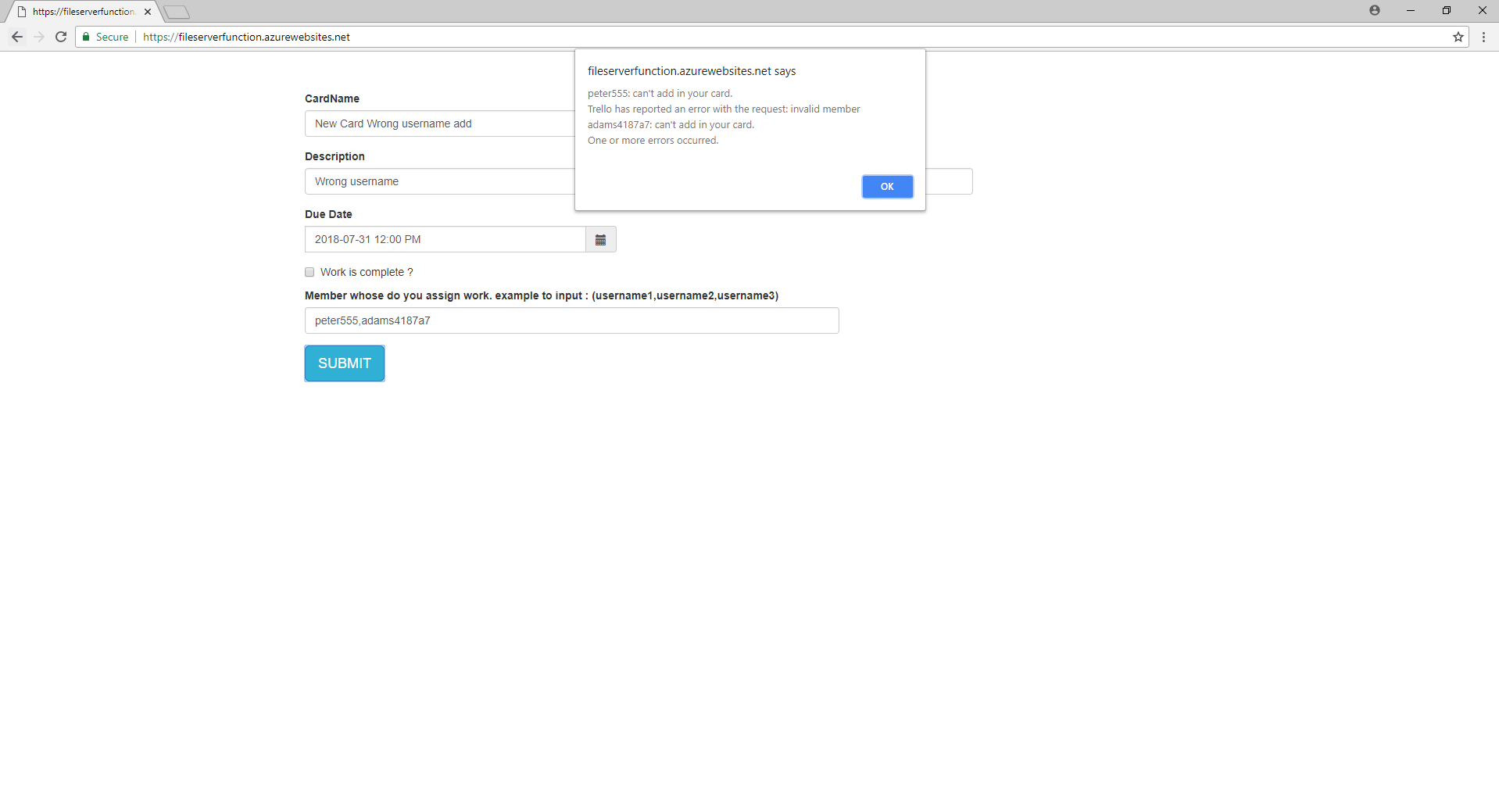
ภาพที่ 53 แสดงตัวอย่างการกรอก input เฉพาะ card name

* 1. **เมื่อกดปุ่ม Submit แล้วข้อมูล card name จะถูกกรอกเข้าไปจะถูกนำไปสร้าง Card ใน Trello ได้สำเร็จ**

ภาพที่ 54 แสดงตัวอย่าง card ใน Trello ตามข้อมูลที่ได้รับ

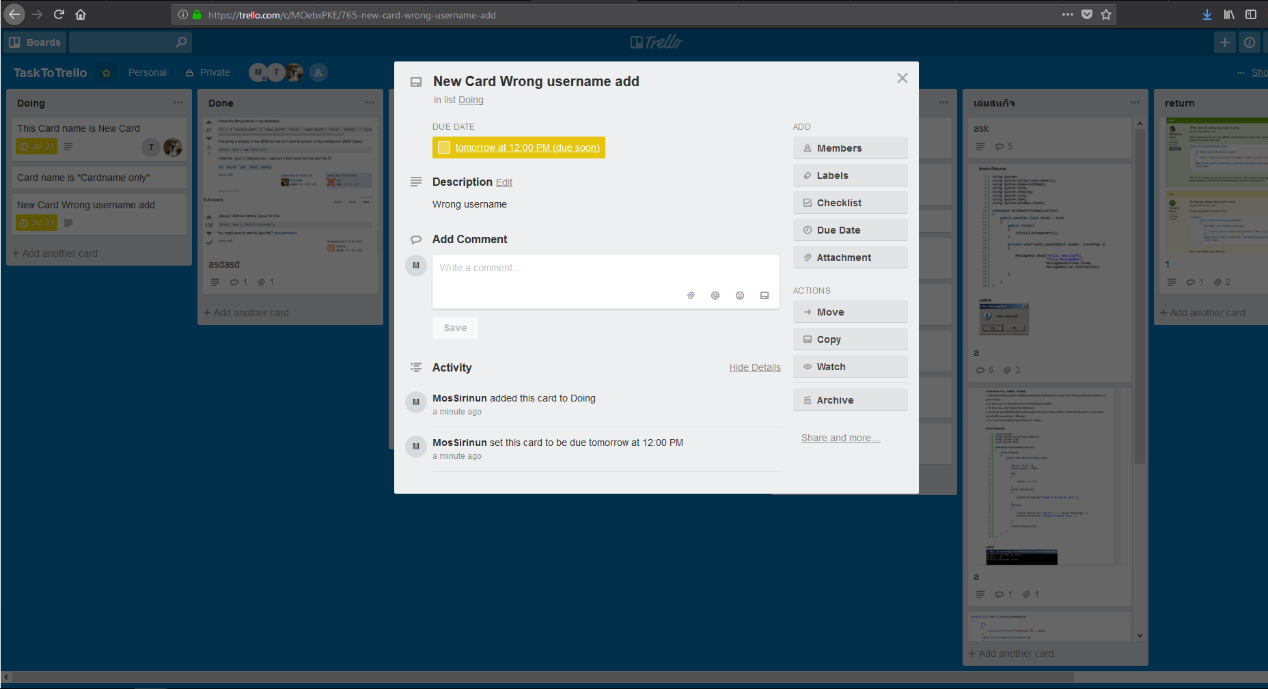
1. สามารถทำการสร้าง card ผ่านหน้า index.html ได้สำเร็จเมื่อกรอกข้อมูลครับยกเว้นusername **โดยปกติแล้วการสร้าง card นั้นจะมีแค่ชื่อของการ์ดก็สามารถสร้างได้สำเร็จ**
   1. **สร้าง card เมื่อทำการกรอกข้อมูลในหน้าฟอร์มครบยกเว้น username ที่ใส่ผิดหรือไม่ได้เป็นสมาชิกภายใน board ของ trello  
      เมื่อกดปุ่ม submit จะมี alert แจ้งเตือนว่า usernameที่กรอกไปจะไม่ถูก เพิ่มเข้าภายใน card**

ภาพที่ 55 แสดงตัวอย่าง Alert ของการกรอก input username ผิด

**เมื่อกดปุ่ม OK ต่อไปก็จะมีการแจ้งเตือนว่า username อะไรบ้างที่ผิดและไม่ถูกแอดเข้าไปภายใน card บ้าง**

ภาพที่ 56 แสดงตัวอย่าง Alert ของการกรอก input username ผิด

* 1. **เมื่อกดปุ่ม Submit แล้วข้อมูลที่จะถูกกรอกเข้าไปจะถูกนำไปสร้าง Card ใน Trello ได้สำเร็จ แต่จะไม่มี username ในการ์ด**

****

ภาพที่ 57 แสดงตัวอย่าง Alert ของการกรอก input username ผิด

# **บทที่ 5 สรุปผลการดำเนินงาน**

## สรุปผลการดำเนินงาน

**ในการพัฒนาเว็บการพัฒนาเว็บแอปพลิเคชัน File Server & Task Management ได้ก่อให้เกิดการพัฒนาทักษะความรู้ในการออกแบบระบบและการพัฒนาระบบ เข้าใจขั้นตอนของการทำงาน ได้เก็บความต้องการของผู้ใช้งาน และการวางแผนก่อนการทำงานเช่น การออกแบบหน้าฟอร์มเพื่อรับค่าข้อมูล เพื่อใช้  
ในการทำงานภายในฟังก์ชันสำหรับการประมวลผล ซึ่งเป็นขั้นตอนสำคัญที่จำเป็นต้องมีในการพัฒนาระบบ และการออกแบบฟังก์ชันเพื่อใช้สำหรับการสร้าง Task ให้ได้ตามที่ต้องการนั้นต้องอาศัยการลองผิดลองถูก  
โดยมีการศึกษาการใช้เครื่องมือและทฤษฎีต่าง ๆ เพื่อเพิ่มประสิทธิภาพความเสถียรภาพของเว็บแอปพลิเคชันให้เกิดประโยชน์สูงสุด และใช้งานได้อย่างมีประสิทธิภาพ**

## ปัญหาและอุปสรรค

* 1. **การวิเคราะห์และออกแบบเป็นไปด้วยความล่าช้า เนื่องจากผู้จัดทำไม่มีประสบการณ์เพียงพอ ต้องศึกษาเพิ่มเติมและของคำแนะนำจากพนักงานที่ปรึกษาบ่อยครั้ง**
  2. **การใช้งานระบบบน Microsoft Azure นั้นแปลกใหม่สำหรับผู้จัดทำจึงทำให้การทำงานในส่วนต่าง ๆ เป็นไปด้วยความยากลำบากต้องมีการศึกษาอยู่ตลอดเวลาและขอคำแนะนำจากพนักงานที่ปรึกษา**
  3. **ในขั้นตอนการทำ Function ต่าง ๆ นั้นจะมี error เกิดขึ้นมากมาย จึงทำให้มีความลำบากในการแก้ปัญหาส่วนนั้น ๆ จึงต้องมีการศึกษาการแก้ปัญหาไปตามแต่ละขั้นตอน**

## ข้อเสนอแนะ

* 1. **ในการออกแบบระบบนั้นควรทำเป็นขั้นตอนให้สำเร็จก่อนเริ่มลงมือทำเพราะจะทำให้ขณะทำงานนั้นต้องมีการปรับเปลี่ยนอยู่ตลอดเวลา**
  2. **การจัดการการส่งข้อมูลภายในฟังก์ชันนั้นค่อนข้างซับซ้อน ควรมีความละเอียดในการทำ**
  3. การทำเมื่อติดปัญหาส่วนไหน ควรรีบปรึกษากับที่ปรึกษาและคนในทีมเพื่อความรวดเร็วของงาน และทำให้งานดำเนินต่อไปได้