เว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนโรคไข้เลือดออกพร้อมวิชวลไลเซชัน

REDUNDANT WEB APPLICATION OF DENGUE VIRUS DATABASE WITH VISUALIZATION

นายดิศวร เนตรศิรินิลกุล

นายวรา ปินตาเสน

นายอาติยะ เอี่ยมเมือง

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

แขนงวิศวกรรมโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ปีการศึกษา 255

เว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนโรคไข้เลือดออกพร้อมวิชวลไลเซชัน

REDUNDANT WEB APPLICATION OF DENGUE VIRUS DATABASE WITH VISUALIZATION

นายดิศวร เนตรศิรินิลกุล

นายวรา ปินตาเสน

นายอาติยะ เอี่ยมเมือง

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

แขนงวิศวกรรมโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ปีการศึกษา 2556

หัวข้อโครงงานวิศวกรรมไฟฟ้า

เรื่อง เว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนโรคไข้เลือดออกพร้อมวิชวลไลเซชัน

โดย นายดิศวร เนตรศิรินิลกุล

นายวรา ปินตาเสน

นายอาติยะ เอี่ยมเมือง

ภาควิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษา อาจารย์กำพล วรดิษฐ์

อาจารย์ที่ปรึกษาร่วม อาจารย์กานต์ เวียรศิลป์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ อนุมัติให้นับโครงงาน วิศวกรรมไฟฟ้า เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.เวคิน ปิยรัตน์)

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ศิริพงษ์ ฉายสินธ์)

..... กรรมการ

(อาจารย์ ธานินทร์ ดวงจันทร์)

..... กรรมการ

(อาจารย์ ดร.กำพล วรดิษฐ์)

..... กรรมการ

(อาจารย์ สุนิศา คุณารักษ์)

เว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนโรคไข้เลือดออกพร้อมวิชวลไลเซชัน

ปีการศึกษา 2556

โดย

นายดิศวร เนตรศิรินิลกุล

นายวรา ปินตาเสน

นายอาติยะ เอี่ยมเมือง

อาจารย์ที่ปรึกษา

อาจารย์กำพล วรดิษฐ์

อาจารย์ที่ปรึกษาร่วม

อาจารย์กานต์ เวียรศิลป์

บทคัดย่อ

โครงงานนี้นำเสนอ เว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนเพื่อการเก็บข้อมูลเชื้อไวรัสไข้เลือดออกที่สามารถเกิดขึ้นได้หลายชนิด หลายรูปแบบ เว็บแอพพลิเคชันนี้สามารถพล็อตกราฟเพื่อแสดงผลการเปรียบเทียบของเชื้อไวรัสแต่ละชนิดออกมาเป็นเปอร์เซ็นต์ได้ โดยเว็บแอพพลิเคชันนี้สร้างขึ้นจากแพลตฟอร์มสองแพลตฟอร์มด้วยกันคือ กูเกิลแอพสคริปท์ (Google apps script) และจังโก้ (Django) ซึ่งแต่ละแพลตฟอร์มจะใช้ผู้สร้างคนละคนกัน จากนั้นจึงนำผลที่ได้มาเปรียบเทียบแพลตฟอร์มทั้งสองในด้านต่าง ๆ เช่น ความสามารถของแต่ละแพลตฟอร์ม ทั้งจากมุมมองของผู้ใช้และผู้สร้าง และเว็บแอพพลิเคชันที่จัดทำขึ้นนี้ จะนำไปใช้งานจริง โดยผู้ใช้ที่เชี่ยวชาญในสาขาที่เกี่ยวข้อง และผู้ที่กำลังศึกษาที่มีความสนใจในเรื่องเชื้อไวรัสไข้เลือดออก

คำสำคัญ : เว็บแอพพลิเคชัน/ ไวรัสไข้เลือดออก/ โรคไข้เลือดออก/ กูเกิลแอพสคริปท์/ จังโก้

REDUNDANT WEB APPLICATION OF DENGUE VIRUS DATABASE WITH VISUALIZATION

Academic Year 2014

By

Mr.Disuan Netsirininkul

Mr.Wara Pintasein

Mr.Artiya Aiemmaung

Advisor

Mr.Kampol Woradit

Co-Advisor

Mr.Kann Vearasilp

Abstract

This project proposes the web-based database application with redundancy for collecting the dengue virus data, which has various types and forms. This web application can plot graphs for comparisons between types of dengue virus as a percentage. The web application was implemented redundantly in two platforms, namely, Google apps script and Django. Each implementation was done by different project member. The obtained results were compared in several aspects, such as, the capacity and the capability from the user and creator perspectives. The proposed web application will be used by experts in the area and anyone, who is interested in dengue virus.

Keywords : Web Application/ Dengue viruses/ Dengue fever/ Google apps script/ Django

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์กำพล วรดิษฐ์ คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมไฟฟ้า-โทรคมนาคม และ อาจารย์กานต์ เวียรศิลป์ คณะเภสัชศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ที่ได้ให้คำแนะนำตลอดจนช่วยตรวจสอบแก้ไขข้อผิดพลาดจนทำให้โครงงานฉบับนี้สำเร็จและขอขอบคุณคณะวิศวกรรมศาสต ร์ที่ได้มอบทุนโครงการทุนนวัตกรรมโครงงานวิศวกรรม ปีการศึกษา 2556 เพื่อสนับสนุนในการทำโครงงานชิ้นนี้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญภาพ	ฉ
บทที่ 1 บทนำ	1
ความเป็นมาของโครงงาน	1
วัตถุประสงค์ของโครงงาน	2
ขอบเขตของโครงงาน	2
ขั้นตอนการค้นคว้าวิจัย	3
ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับโครงงาน	4
Google apps script	4
Gmail	14
Google drive	19
MicroRNA (miRNA)	27
Model View Controller (MVC)	30
Django	30
บทที่ 3 ขั้นตอนการสร้างและวิธีการสร้าง	34
ขั้นตอนการจัดการไฟล์ฐานข้อมูลด้วยภาษาไพธอน (Python)	34
วิธีการสร้างเว็บแอพพลิเคชันด้วยจังโก้ (Django)	37

สารบัญ (ต่อ)

	หน้าที่
วิธีสร้างเว็บแอพพลิเคชันด้วย Google apps script	47
บทที่ 4 ผลการทดลอง	64
ผลการเปรียบเทียบ Google apps script กับ Django	64
ความสามารถของเว็บแอพพลิเคชัน	69
ส่วนของแพลตฟอร์มจังโก้	70
ส่วนของแพลตฟอร์ม Google apps script	77
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	80
สรุปผลการดำเนินงาน	80
ปัญหาการดำเนินงาน	81
ข้อเสนอแนะสำหรับการพัฒนาโครงงาน	81
เอกสารอ้างอิง	82
ภาคผนวก	84

สารบัญภาพ

รูปที่	หน้าที่
2.1 การเรียกใช้ Google apps script เมื่อ Google drive ไม่ปรากฏให้ใช้งาน	5
2.2 เชื่อมต่อ Google apps script ให้ใช้งานบน Google drive	6
2.3 การเรียกใช้งาน Google apps script บน Google drive	7
2.4 การสร้าง Google apps script	8
2.5 หน้าต่างของ Google apps script	9
2.6 ตัวอย่างไฟล์ Script ที่ถูกสร้างขึ้น	10
2.7 การ Deploy as web app	11
2.8 ตั้งค่าเว็บแอพพลิเคชัน	12
2.9 URL ของเว็บแอพพลิเคชัน	13
2.10 http://accounts.google.com เว็บเพจ	14
2.11 การสมัครใช้งาน Gmail (1)	15
2.12 การสมัครใช้งาน Gmail (2)	16
2.13 เพิ่มรูปภาพให้กับโปรไฟล์ของ Gmail	17
2.14 ยืนยันเพื่อเริ่มต้นการใช้งาน Gmail	17
2.15 ล็อคอินเข้าใช้งาน Google apps	18
2.16 ล็อคอิน Google account สำเร็จ	19
2.17 การเข้า Google drive ด้วย Gmail	21
2.18 การสร้าง Spreadsheet บน Google drive	22
2.19 Google spreadsheet	23
2.20 การ Import file ลงบน Google spreadsheet	24
2.21 ไฟล์ที่ถูก Import ลงบน Google drive	25
2.22 ข้อมูลใน Google spreadsheet ที่ได้จากการ Import	26
2.23 สายรหัสทางพันธุกรรม miRNA	27

สารบัญภาพ (ต่อ)

รูปที่	หน้าที่
2.24 การเกิดมิวเทชัน (Mutation)	28
2.25 การเกิดอินเซริสชัน (Insertion)	28
2.26 การเกิดดีลีทชัน (Deletion)	29
2.27 การเปลี่ยนแปลงของรหัสทางพันธุกรรม	29
3.1 คำสั่งสร้างโปรเจค	37
3.2 ไฟล์ในโฟลเดอร์โปรเจค	37
3.3 คำสั่งรันเซิร์ฟเวอร์	38
3.4 รันคำสั่งรันเซิร์ฟเวอร์	38
3.5 เซิร์ฟเวอร์สามารถใช้งานได้	39
3.6 คำสั่งสตาร์ทแอพ plot	39
3.7 โฟลเดอร์ plot	40
3.8 ภายในโฟลเดอร์ plot	40
3.9 เปิดโปรแกรม MySQL	41
3.10 รันคำสั่งสร้าง Database ใน MySQL	41
3.11 ไฟล์ settings.py	42
3.12 ใส่ชื่อฐานข้อมูลและพาสเวิร์ด	42
3.13 ใส่ชื่อแอพ plot ที่สร้างขึ้น	43
3.14 คำสั่ง syncdb	43
3.15 รันคำสั่ง syncdb	44
3.16 สร้างไฟล์ tabplot.py	44
3.17 โฟลเดอร์ templates	45
3.18 โฟลเดอร์ plot	45
3.19 ตั้งค่า TEMPLATE_DIRS	46

สารบัญภาพ (ต่อ)

รูปที่	หน้าที่
3.20 ไฟล์ html และ css	46
3.21 ตั้งค่า STATICFILES_DIRS	47
3.22 ล็อคอินเข้าใช้งาน Google apps	47
3.23 ล็อคอินเข้าใช้งาน Google apps สำเร็จ	48
3.24 การเข้าใช้งาน Google drive	48
3.25 สร้าง Spreadsheet ใน Google apps	49
3.26 Import file ลงบน Google spreadsheet (1)	50
3.27 Import file ลงบน Google spreadsheet (2)	51
3.28 ไฟล์ที่ Import ลงบน Google drive	52
3.29 ข้อมูลใน Google spreadsheet	53
3.30 การจัดการไฟล์ฐานข้อมูลบน Google drive	54
3.31 การเรียกใช้ Google apps script เมื่อ Google drive ไม่ปรากฏการใช้งาน	55
3.32 เชื่อมต่อ Google apps script บน Google drive	56
3.33 การใช้งาน Google apps script บน Google drive	57
3.34 การสร้าง Google apps script	58
3.35 โปรแกรม Google apps script	59
3.36 ไฟล์ Script ที่สร้างขึ้น	60
3.37 Deploy as web app	61
3.38 การตั้งค่าเว็บแอพพลิเคชัน	62
3.39 URL เว็บแอพพลิเคชัน	63
4.1 การสำรองข้อมูลบน Google apps script	64
4.2 การประมวลผลกราฟของ Google apps script	65
4.3 การแสดงกราฟของ Google apps script	66

สารบัญภาพ (ต่อ)

รูปที่	หน้าที่
4.4 การแสดงกราฟของ Django	67
4.5 ความจุข้อมูลของ Google apps script	68
4.6 ความจุข้อมูลของ Django	69
4.7 หน้า Home	70
4.8 Dengue Fever Menu	71
4.9 หน้า Dengue Fever > Ind3	71
4.10 หน้า Dengue Hemorrhagic Fever > Ind1	72
4.11 การเลือกระหว่าง Frequency และ Percentage	72
4.12 ช่องระบุตำแหน่งและปุ่ม Structural Proteins	73
4.13 การระบุตำแหน่งของกราฟ	74
4.14 แสดงข้อความเตือนเมื่อผู้ใช้ระบุตำแหน่งเริ่มต้น มากกว่าตำแหน่งสุดท้าย	74
4.15 แสดงกราฟตำแหน่งที่ 95 ถึง 436	75
4.16 หน้าต่างบันทึกกราฟเป็นไฟล์ .PNG	76
4.17 หน้าต่างบันทึกข้อมูลบนการเป็นไฟล์ .CSV	76
4.18 ปุ่มลิงค์ไป Google apps script	77
4.19 วิธีใช้เว็บแอพพลิเคชันฐานข้อมูล Dengue virus โดย Google apps script	78
4.20 วิธีใช้เว็บแอพพลิเคชันบน Google apps script	78
4.21 การแสดงกราฟของ Google apps script	79

บทที่ 1

บทนำ

1. ความเป็นมาของโครงงาน

ในสถานการณ์ปัจจุบันการระบาดของโรคไข้เลือดออกในประเทศไทยยังเป็นหนึ่งในปัญหาด้านสุขภาพของประชาชนตามชนบท จากค่าสถิติข้อมูลโรคไข้เลือดออกจากกระทรวงสาธารณสุขได้บันทึกไว้ว่ามีผู้ป่วยด้วยโรคไข้เลือดออกเป็นจำนวน 42,244 ราย และมีการเสียชีวิตทั้งหมด 44 ราย(ข้อมูลวันที่ 25 กันยายน พ.ศ. 2555 "รายงานสถานการณ์ไข้เลือดออกประจำปี 2012") คิดเป็นอัตราผู้ป่วยประมาณ 70 รายต่อประชากรหนึ่งแสนคนและมีอัตราการเสียชีวิตประมาณ 1 รายต่อในผู้ป่วยโรคไข้เลือดออกหนึ่งหมื่นรายจากข้อเท็จจริงดังกล่าวทำให้เชื่อได้ว่าโรคไข้เลือดออกยังเป็นปัญหาที่สำคัญของประเทศไทย

แต่อย่างไรก็ตาม สืบเนื่องจากการเก็บข้อมูลทางคลินิคของหลายมหาวิทยาลัยแพทย์ชั้นนำภายในประเทศไทย ส่งผลให้ประเทศไทยได้ถูกนับว่าเป็นหนึ่งในประเทศที่มีความก้าวหน้าในหัวข้อวิจัยไวรัสไข้เลือดออก ซึ่งที่ผ่านมาในระยะสิบปีล่าสุด หลายๆสถาบันวิจัยในประเทศไทยได้มีการตีพิมพ์งานวิจัยลงในวารสารวิชาการชั้นนำในระดับนานาชาติอย่างต่อเนื่องเพียงแต่น่าเสียดายว่าประเทศไทยยังไม่มีระบบการจัดข้อมูลโรคไข้เลือดออกโดยอาศัยระบบชีวสารสนเทศอย่างมีระบบ ในการศึกษาวิจัยในระบบบูรณาการนั้นสิ่งที่จำเป็นต่อการทำงานร่วมกันระหว่างนักวิทยาศาสตร์ชั้นนำหลายสถาบันวิจัยก็คือระบบฐานข้อมูลที่เชื่อถือได้ และอำนวยความสะดวกแก่นักวิจัยที่มาจากต่างสาขา

ดังนั้นโครงงานวิจัยนี้จึงต้องการศึกษาและมีวัตถุประสงค์เพื่อพัฒนาระบบชีวสารสนเทศเพื่อทำการเก็บรวบรวมข้อมูลทางอณูชีววิทยาและเชื้อไวรัสไข้เลือดออกโดยเฉพาะและข้อมูลจีโนมของทั้งคนไข้และเชื้อไวรัสไข้เลือดออก สายพันธุ์ที่ถูกพัฒนาในห้องวิจัยของประเทศไทยข้อมูลสภาวะ Quasispecies ของตัวเชื้อไวรัสทั้งนี้เพื่อการจัดข้อมูลให้เป็นระเบียบและเปิดช่องทนำเสนอข้อมูลที่เข้าถึงง่ายและเป็นระบบแก่นักวิจัยทั่วไป นอกจากนี้ยังมีวัตถุประสงค์ที่จะศึกษาความหมาย

หลากหลายของ Quasispecies ในเชื้อไวรัสไข้เลือกออกเพื่อการค้นหาทางกลไกการเปลี่ยนแปลงระบบพันธุกรรมของไวรัสเมื่อผู้ป่วยมีการติดเชื้อ เพื่อที่จะสามารถบ่งชี้ Genetic Marker ของตัวไวรัสเองโดยมีความคาดหวังที่จะสามารถคัดแยกผู้ป่วยที่มีความเสี่ยงต่อการเกิดอาการ Dengue Hemorrhagic Fever ออกจาก Dengue Fever ได้

อนึ่งการนำเสนอฐานข้อมูลชีวสารสนเทศของโครงงานวิจัยนี้จะทำในรูปแบบเว็บไซต์เป็น frontend และอาศัย Scripting และฐานข้อมูล MySQL เป็น backend ซึ่งแพลตฟอร์มที่เลือกใช้ทำโครงงานคือ Django กับ Google apps script โดยที่แต่ละแพลตฟอร์มจะใช้ผู้สร้างคนละคนกัน จากนั้นนำผลลัพธ์ที่ได้มาเปรียบเทียบเพื่อหาจุดบกพร่องข้อได้เปรียบเสียเปรียบของทั้งสองแพลตฟอร์มทั้งจากมุมมองของผู้ใช้และผู้สร้าง

2. วัตถุประสงค์ของโครงงาน

2.1 สร้างเว็บแอพพลิเคชันฐานข้อมูลที่ใช้รับข้อมูล miRNA จัดเก็บไว้ในฐานข้อมูลและสามารถไล่ดูข้อมูลโดยแสดงผลลัพธ์ออกมาในรูปของกราฟ

2.2 เปรียบเทียบแพลตฟอร์มระหว่าง Google apps script กับ Django ว่าแพลตฟอร์มไหนมีประสิทธิภาพกว่ากันทั้งในมุมมองของผู้ใช้และผู้สร้าง โดยเปรียบเทียบจาก

2.2.1 ความสามารถของแต่ละแพลตฟอร์ม

2.2.2 ความยากง่ายในการปรับปรุงแก้ไขข้อผิดพลาด

2.2.3 ความจุที่สามารถรองรับข้อมูลได้

3. ขอบเขตของโครงงาน

3.1. สร้างเว็บแอพพลิเคชันฐานข้อมูล miRNA ประกอบด้วยส่วนต่างๆดังนี้

3.1.1 ส่วนจัดเก็บข้อมูลลงบนฐานข้อมูลสำหรับผู้ดูแลระบบ

3.1.2 ส่วนประมวลผลข้อมูลในฐานข้อมูล

3.1.3 ส่วนแสดงข้อมูลจากฐานข้อมูล

3.1.4 ส่วนแสดงเนตเวิร์คไดอะแกรม เช่น กราฟ

3.2. เปรียบเทียบแพลตฟอร์มระหว่าง Django กับ Google apps script ดังนี้

3.2.1 ความสามารถของแต่ละแพลตฟอร์ม

3.2.2 ความรวดเร็วในการประมวลผลของแต่ละแพลตฟอร์ม

3.2.3 ความจุที่สามารถรองรับข้อมูลได้

3.2.4 ความสามารถในการแสดงข้อมูลในรูปของกราฟ

4. ขั้นตอนการค้นคว้าวิจัย

4.1 ดาวน์โหลดติดตั้งซอร์ฟแวร์ที่ใช้และศึกษาวิธีการใช้งาน

4.2 เขียนโปรแกรมจัดการไฟล์ฐานข้อมูลให้เป็นไฟล์ Fasta ไฟล์ XML และไฟล์ CSV

4.3 ศึกษาการสร้างเว็บแอพพลิเคชันฐานข้อมูลของทั้งสองแพลตฟอร์ม

4.4 ออกแบบฐานข้อมูลและหน้าตาของเว็บแอพพลิเคชัน

4.5 สร้างเว็บแอพพลิเคชันทั้งสองแพลตฟอร์ม

4.6 ทดสอบเว็บแอพพลิเคชันและปรับปรุงแก้ไขจุดบกพร่อง

4.7 สรุปผล

5. ประโยชน์ที่คาดว่าจะได้รับ

5.1 ผลเปรียบเทียบแพลตฟอร์มที่ได้จะใช้ตัดสินใจในการเลือกใช้แพลตฟอร์มในการสร้างเว็บแอพพลิเคชันที่ตอบสนองตามความต้องการของทั้งผู้สร้างและผู้ใช้ได้ในอนาคต

5.2 ผลลัพธ์เว็บแอพพลิเคชันที่ได้จะได้รับการนำไปใช้งานจริงโดยโดยผู้ใช้ที่เชี่ยวชาญในสาขาที่เกี่ยวข้อง และผู้ที่กำลังศึกษาที่มีความสนใจในเรื่องเชื้อไวรัสไข้เลือดออก

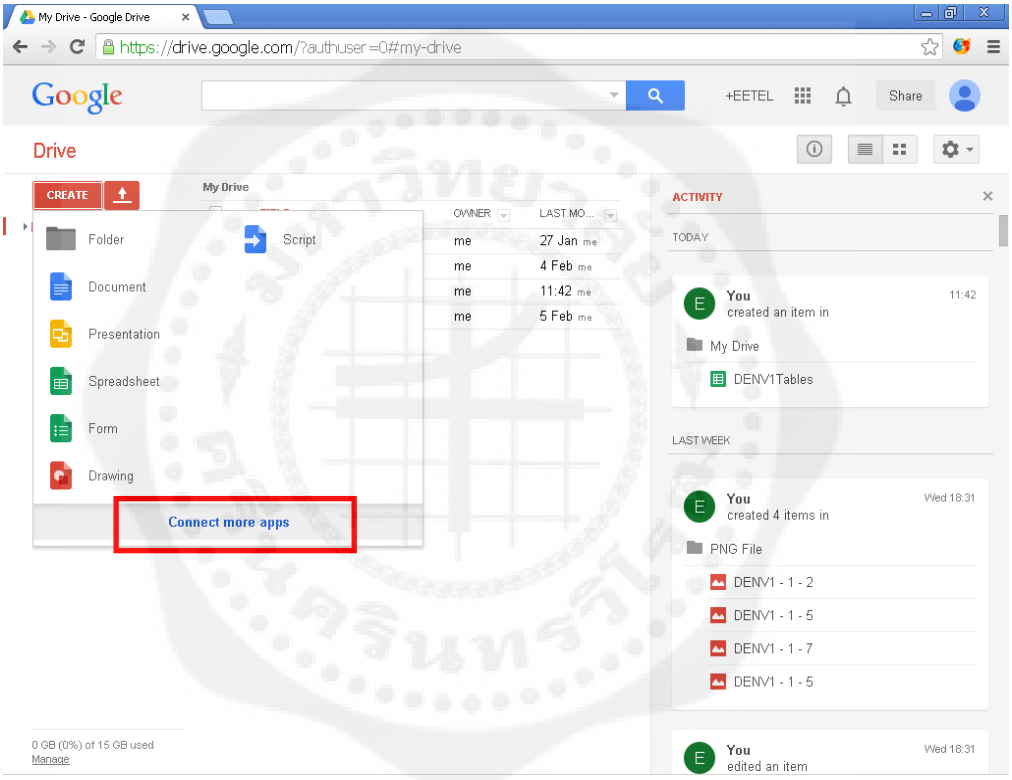
บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับโครงงาน

1. Google apps script เป็นโปรแกรมสคริปต์ภาษาของ Google ที่ใช้แบบภาษาของ Javascript เป็นหนึ่งในงานบริการเสริมของ Google apps ช่วยในการใช้งานในการจัดการงานบริการของ Google apps ไม่ว่าจะเป็น Google mail (Gmail) , Google drive , Google maps และ Google apps อื่นๆ อีกมากมายได้ทั้งหมด ซึ่งในการใช้งาน Google apps script ไม่จำเป็นต้องมีเซิร์ฟเวอร์ ซอฟต์แวร์เว็บเซิร์ฟเวอร์ หรือซอฟต์แวร์ฐานข้อมูล และไม่จำเป็นต้องมีการบำรุงรักษาเซิร์ฟเวอร์ เช่น การสำรองข้อมูล การสำรองเซิร์ฟเวอร์ และผู้สร้างเว็บแอพพลิเคชันสามารถแก้ไขโปรแกรมได้จากคอมพิวเตอร์เครื่องใดก็ได้ ที่มีการเชื่อมต่ออินเทอร์เน็ต

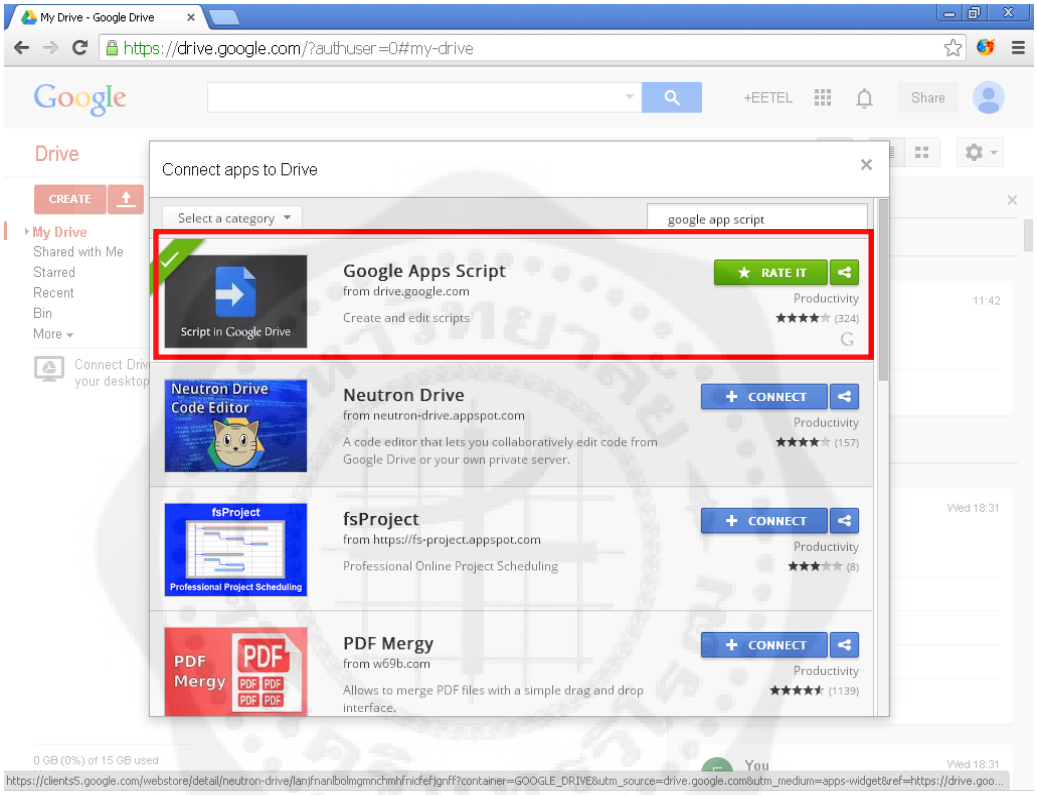
1.1 การใช้งาน Google apps script

1.1.1 เมื่อเรามี Google account (Gmail) แล้วในส่วนของ Google drive ที่ปุ่มสร้าง (Create) จะ มีบริการ Google apps ต่างๆ ซึ่งตัว Google apps script นั้นเราสามารถเรียกใช้งานได้จากตรงนี้ในกรณีใช้งานครั้งแรกบน Google drive อาจจะไม่ปรากฏ Google apps script ให้ใช้งานต้องเรียกใช้งานโดยให้คลิกที่ปุ่ม Create และคลิกที่ Connect more apps ดังรูปที่ 2.1



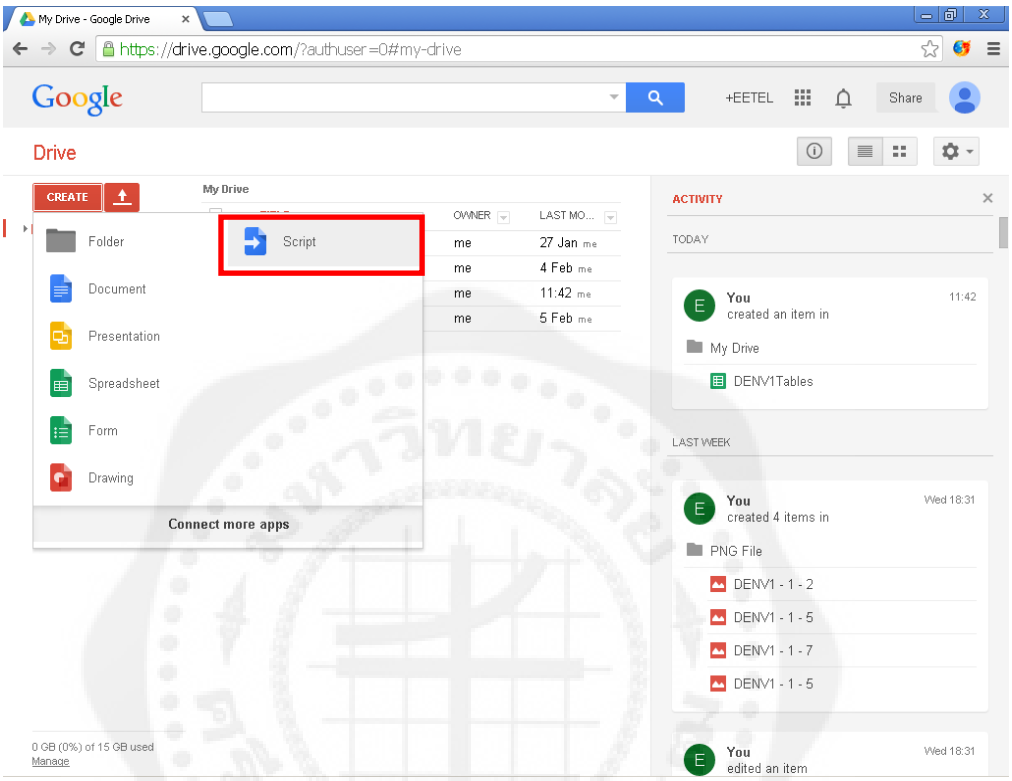
รูปที่ 2.1 การเรียกใช้ Google apps script เมื่อ Google drive ไม่ปรากฏให้ใช้งาน

1.1.2 ในส่วนนี้มีจะมีบริการของ Google apps มากมาย ให้เราหา Google apps script ตรงช่อง Search apps แล้วให้ คลิกปุ่ม Connect พอเชื่อมต่อสำเร็จตรงปุ่ม Connect นั้นจะเปลี่ยนเป็นคำว่า Rate it เราก็จะได้ Google apps script มาใช้งานบน Google drive ของเราเอง ดังรูปที่ 2.2



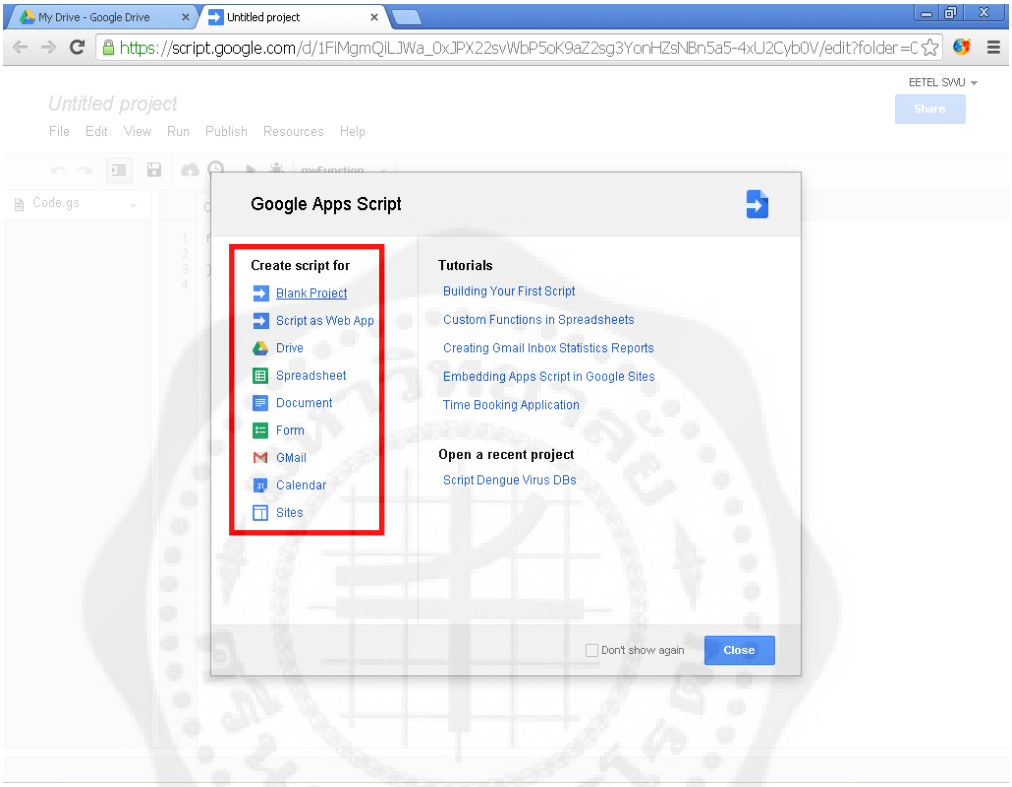
รูปที่ 2.2 เชื่อมต่อ Google apps script ให้ใช้งานบน Google drive

1.1.3 ในการสร้าง Google apps script ทำได้โดยคลิกปุ่ม Create และเลือก Script ดังรูปที่ 2.3



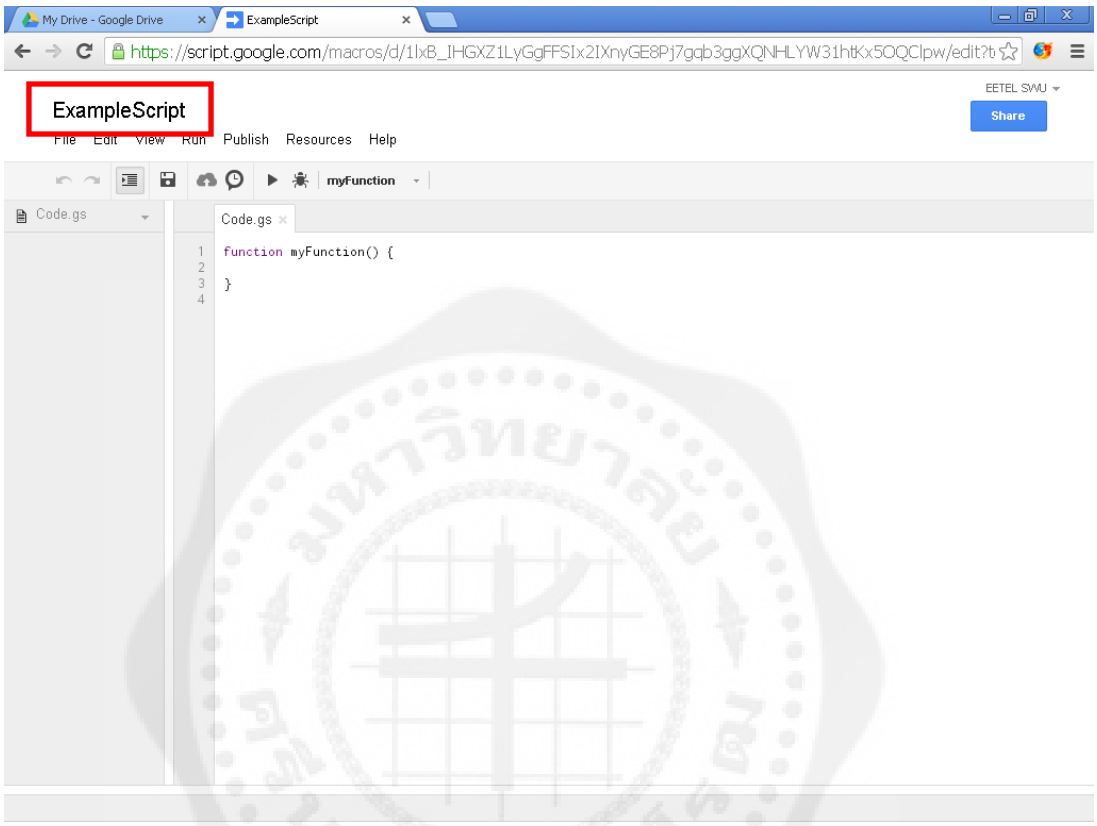
รูปที่ 2.3 การเรียกใช้งาน Google apps script บน Google drive

1.1.4 จะปรากฏหน้าต่าง Google apps script ขึ้นมา ตรงส่วน Create script for จะเป็นตัวเลือกในการสร้าง Script แบบต่างๆ ตัวอย่างเช่นเราจะสร้าง script เปล่าๆ ขึ้นมา 1 ไฟล์ ให้เลือกในส่วน Create script for เป็น Blank Project ดังรูปที่ 2.4

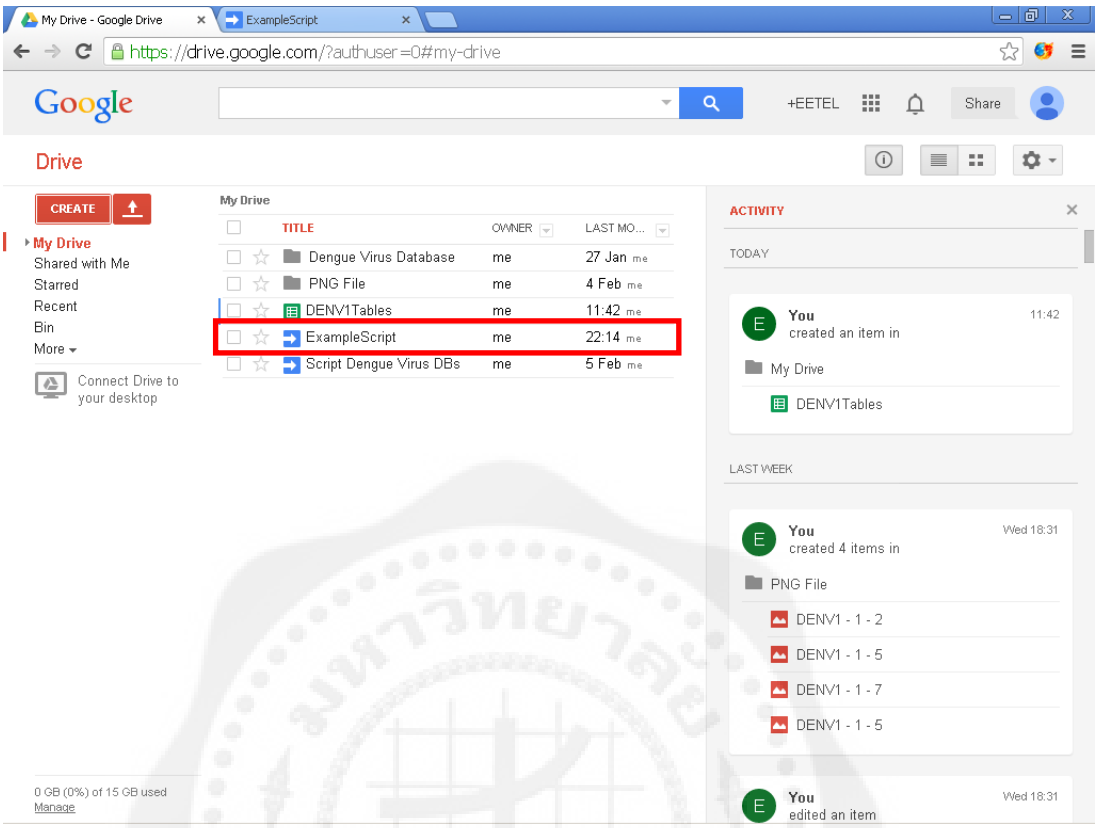


รูปที่ 2.4 การสร้าง Google apps script

1.1.5 จะขึ้นหน้าต่าง script ทำการตั้งชื่อตรง Untitled Project ก็จะได้ไฟล์ Script มาใช้งาน เช่น ตั้งชื่อเป็น ExampleScript ดังรูปที่ 2.5 และรูปที่ 2.6



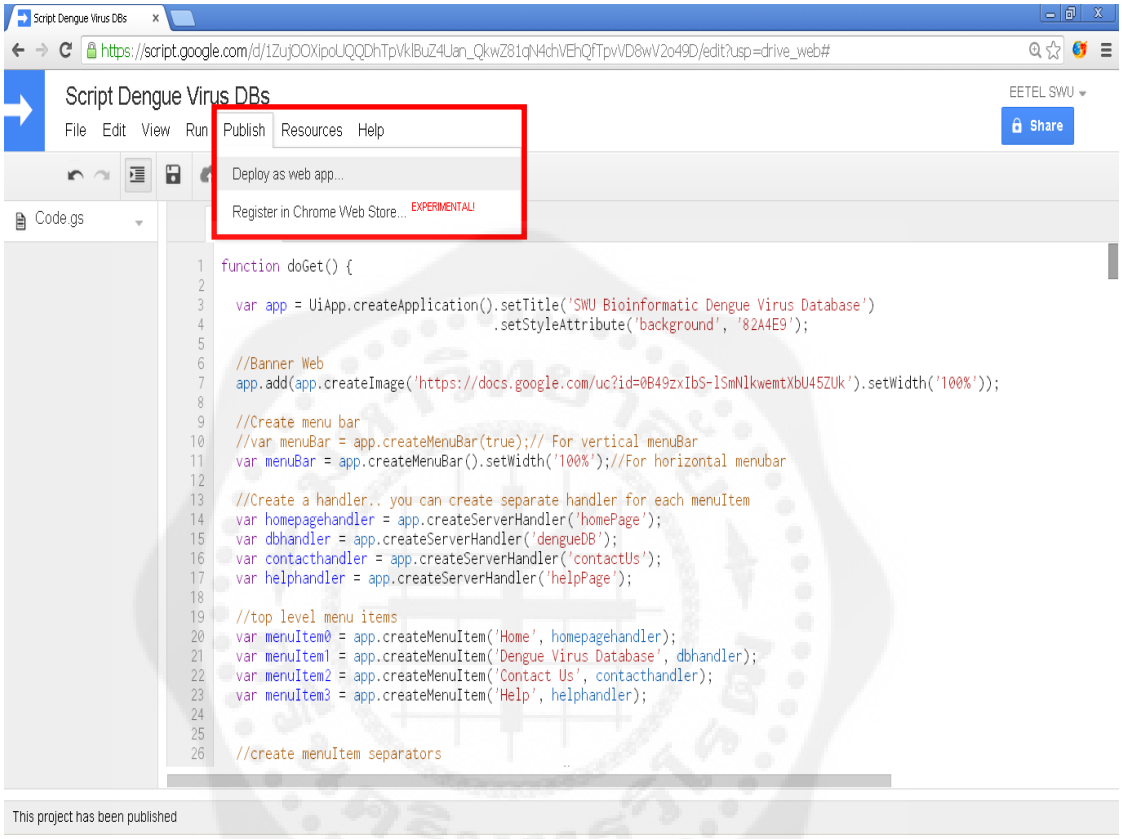
รูปที่ 2.5 หน้าต่างของ Google apps script



รูปที่ 2.6 ตัวอย่างไฟล์ Script ที่ถูกสร้างขึ้น

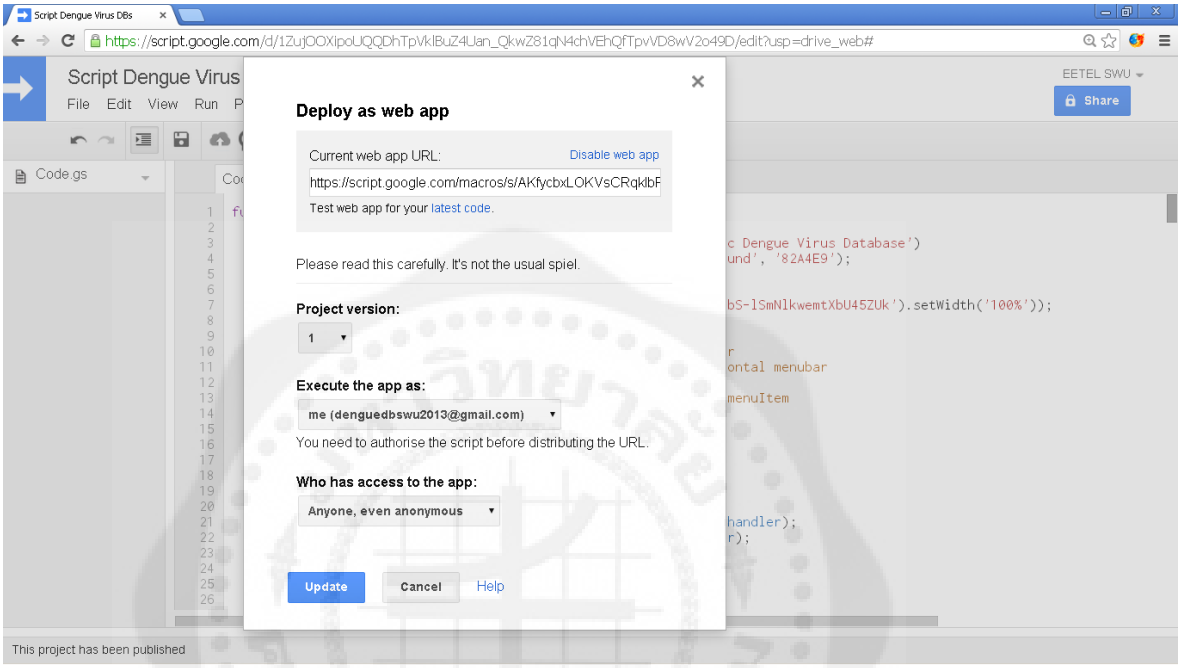
1.2 การ Deploy เป็นเว็บแอพพลิเคชันของ Google apps script

1.2.1 เปิด Google apps script ขึ้นมา ให้คลิกที่แถบเมนูชื่อว่า Publish และ Deploy as web app… ดังรูปที่ 2.7



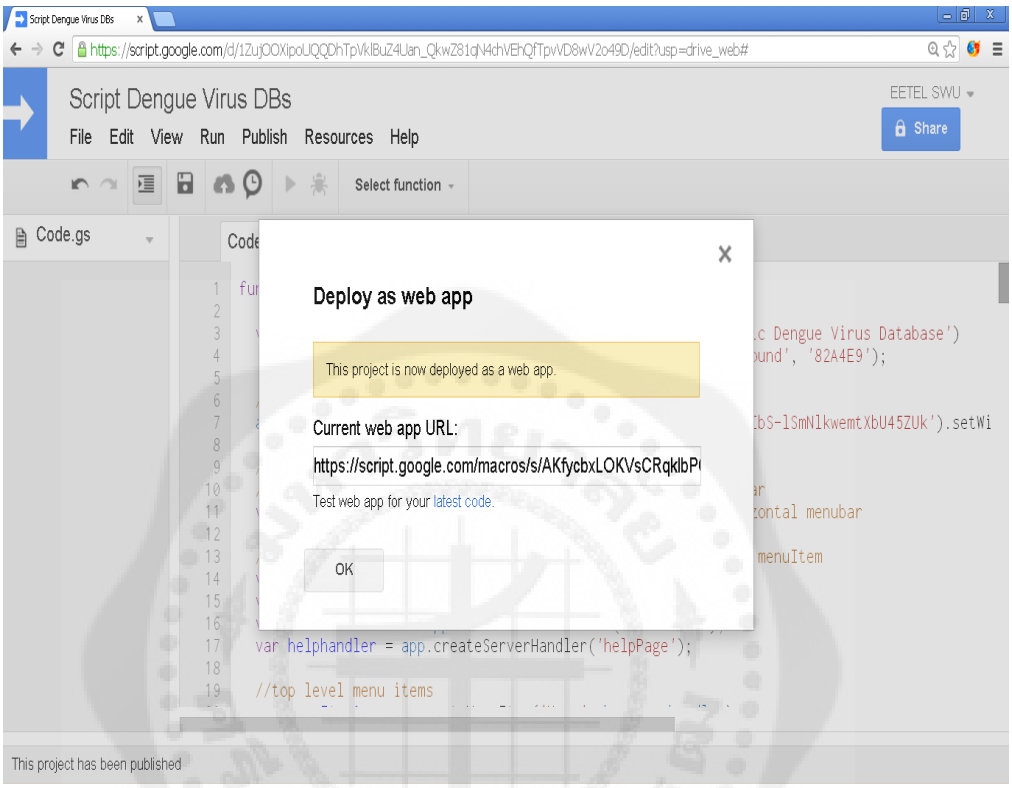
รูปที่ 2.7 การ Deploy as web app

1.2.2 จะปรากฏหน้าต่างดังรูปที่ 2.8 ให้ตั้งค่าดังนี้ ตรงส่วน Project version: ให้คลิก Manage version และตรงส่วนที่เป็น Who has access to the app: ให้ตั้งค่าเป็น Anyone, even anonymous เพื่อให้คนนอกเข้ามาใช้งาน URL ของเราได้ แล้วคลิก Update



รูปที่ 2.8 ตั้งค่าเว็บแอพพลิเคชัน

1.2.3 จะปรากฏหน้าต่างดังรูปที่ 2.9 เราก็จะได้ URL ของเว็บแอพพลิเคชันเราและเว็บแอพพลิเคชันที่เราสร้างขึ้นด้วย Google apps script

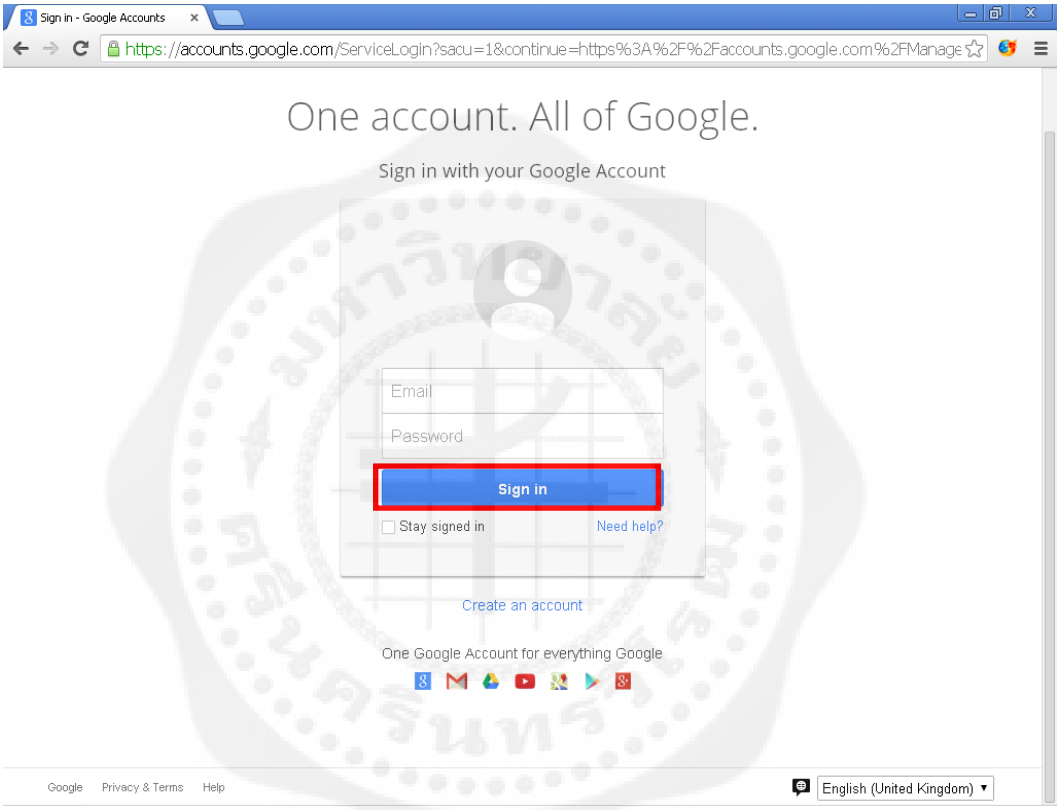


รูปที่ 2.9 URL ของเว็บแอพพลิเคชัน

2. Gmail คือ บริการฟรีอีเมล์ภายใต้การบริหารของ Google สำหรับผู้ที่ต้องการใช้บริการของ Google apps ต้องมี Gmail เท่านั้นถึงจะใช้บริการได้

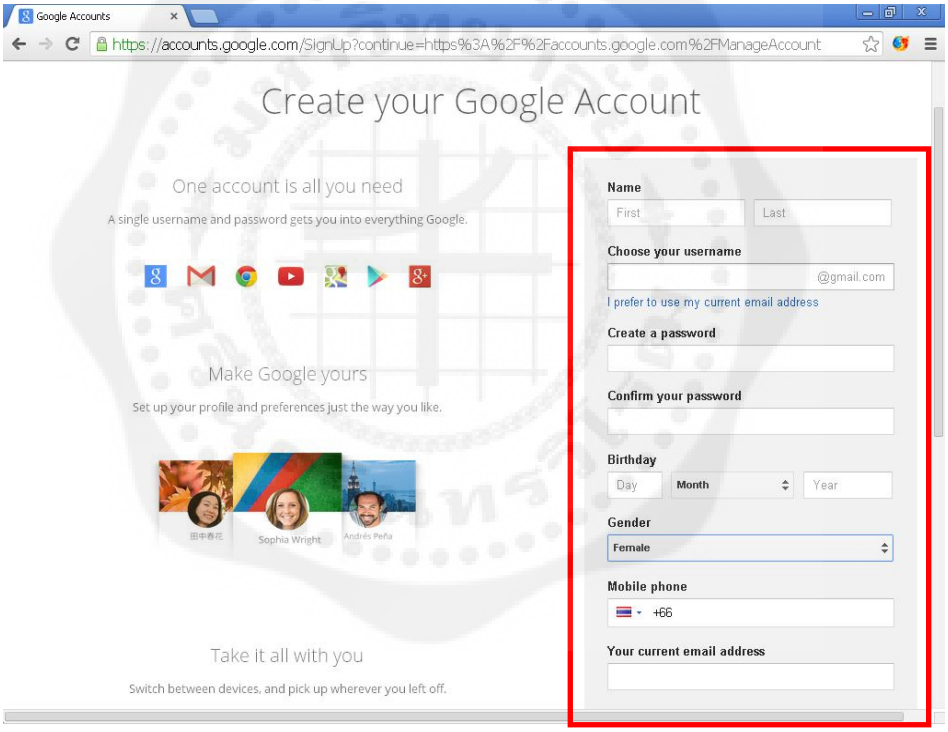
2.1 ขั้นตอนการสมัคร

2.1.1 ให้ทำการเข้าไปที่เว็บไซต์ http://accounts.google.com และคลิกที่ Create an account ดังรูปที่ 2.10



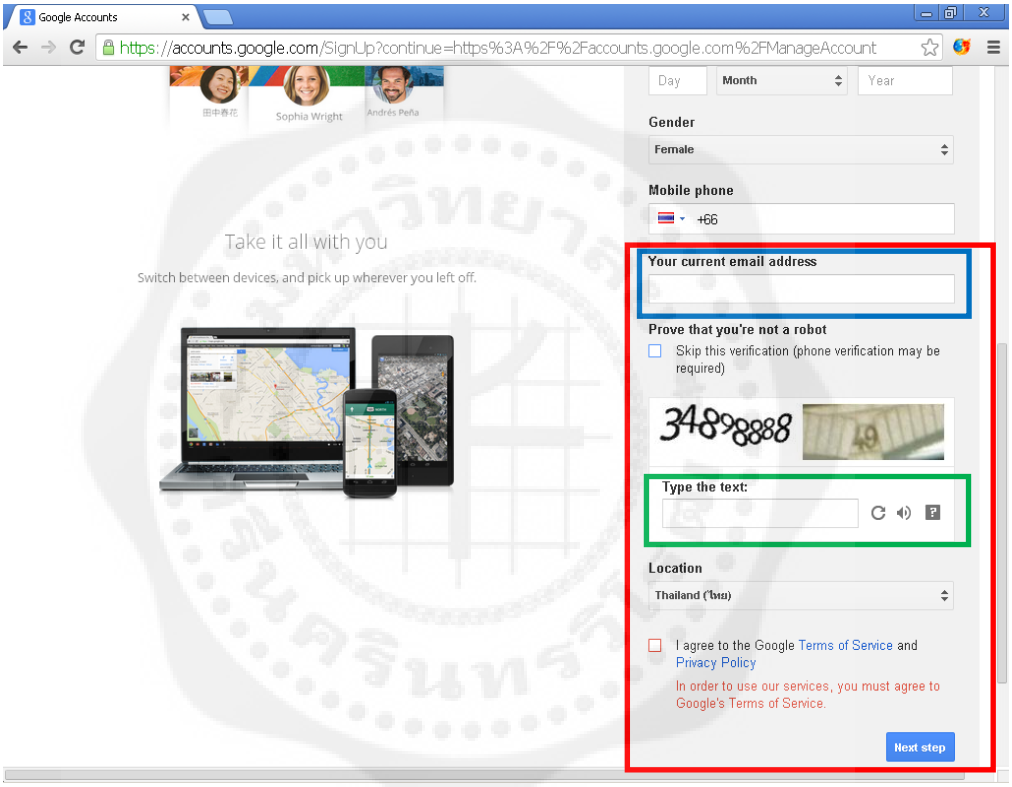
รูปที่ 2.10 http://accounts.google.com เว็บเพจ

2.1.2 เมื่อเข้ามาในส่วนของ Create an account ให้ทำการกรอกข้อมูลการสมัคร ซึ่งจะมีช่องกรอกข้อมูลทางขวาของเว็บเพจ ซึ่งได้แก่ 1.ช่องสำหรับกรอกชื่อ - นามสกุล 2.ช่องใส่ชื่อผู้ใช้หรือชื่ออีเมล์ อาทิเช่น Gunoob@gmail.com ในการกรอกขื่อผู้ใช้ ไม่ต้องกรอก @gmail.com ระบบจะทำการกรอกให้โดยอัตโนมัติ 3.ช่องสำหรับการสร้างหรือใส่รหัสผ่านของอีเมล์ 4.ช่องสำหรับให้เราใส่รหัสผ่านอีเมล์อีกครั้ง เพื่อยืนยัน 5.ช่องสำหรับให้เราใส่วันที่เกิด (วันที่/เดือน/ปี ค.ศ.) 6.ช่องสำหรับให้เราเลือกเพศ (หญิง ชาย และ อื่นๆ) 7.ช่องสำหรับให้เราทำการกรอก โทรศัพท์มือถือ โดยให้ +66 แทนเป็นเลข 0 ตัวแรกของเบอร์ เช่น เบอร์0842231xxx ก็จะเป็น +66842231xxx เป็นต้นและ 8.ช่องสำหรับให้เราทำการใส่อีเมล์เดิมหรืออีเมล์ปัจจุบันที่เรามี ดังรูปที่ 2.11



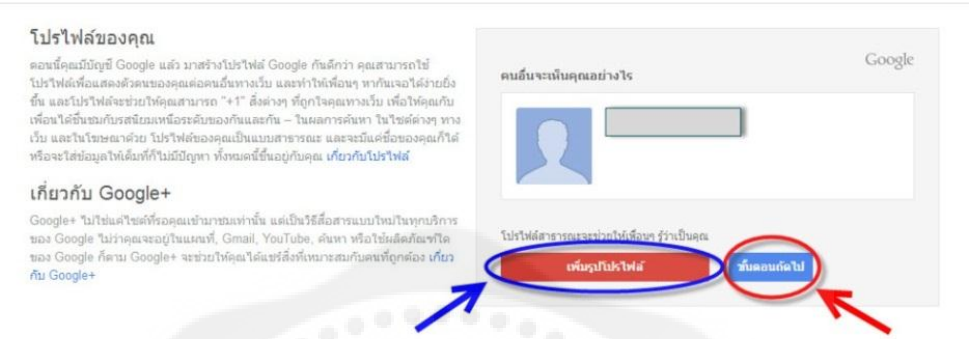
รูปที่ 2.11 การสมัครใช้งาน Gmail (1)

2.1.3 เมื่อทำการกรอกข้อมูลเรียบร้อยแล้วถัดลงมาจะเป็นช่องการพิสูจน์ว่าไม่ได้เป็นหุ่นยนต์ โดยถ้าเราติ๊กในช่องสีน้ำเงินจะเป็นการยืนยันทางโทรศัพท์ ส่วนถ้าไม่ติ๊กในช่องสีน้ำเงินก็จะเป็นการยืนยันโดยการกรอกอักษรจากในภาพลงในช่องสำหรับกรอก แนะนำให้ยินยันแบบกรอกตัวอักษรตามภาพจะสะดวกกว่า และสุดท้ายเป็นช่องการเลือกตำแหน่งที่อยู่ ให้ทำการเลือก ไทย จากนั้นทำการติ๊กในวงกลมสีเขียว เพื่อยอมรับเงื่อนไขต่างๆ และทำการกดที่ปุ่มล่างสุดสีฟ้าขั้นตอนถัดไป (Next Step) ดังรูปที่ 2.12



รูปที่ 2.12 การสมัครใช้งาน Gmail (2)

2.1.4 เมื่อกดขั้นตอนถัดไปแล้ว ก็จะมาหน้าต่างนี้ ถ้าต้องการจะเพิ่มรูปภาพโปรไฟล์ ให้ทำการคลิกที่วงกลมสีน้ำเงิน ถ้าต้องการแก้ไขในภายหลังให้ทำการ คลิกที่ขั้นตอนถัดไป วงกลมสีแดงได้เลย ดังรูปที่ 2.13

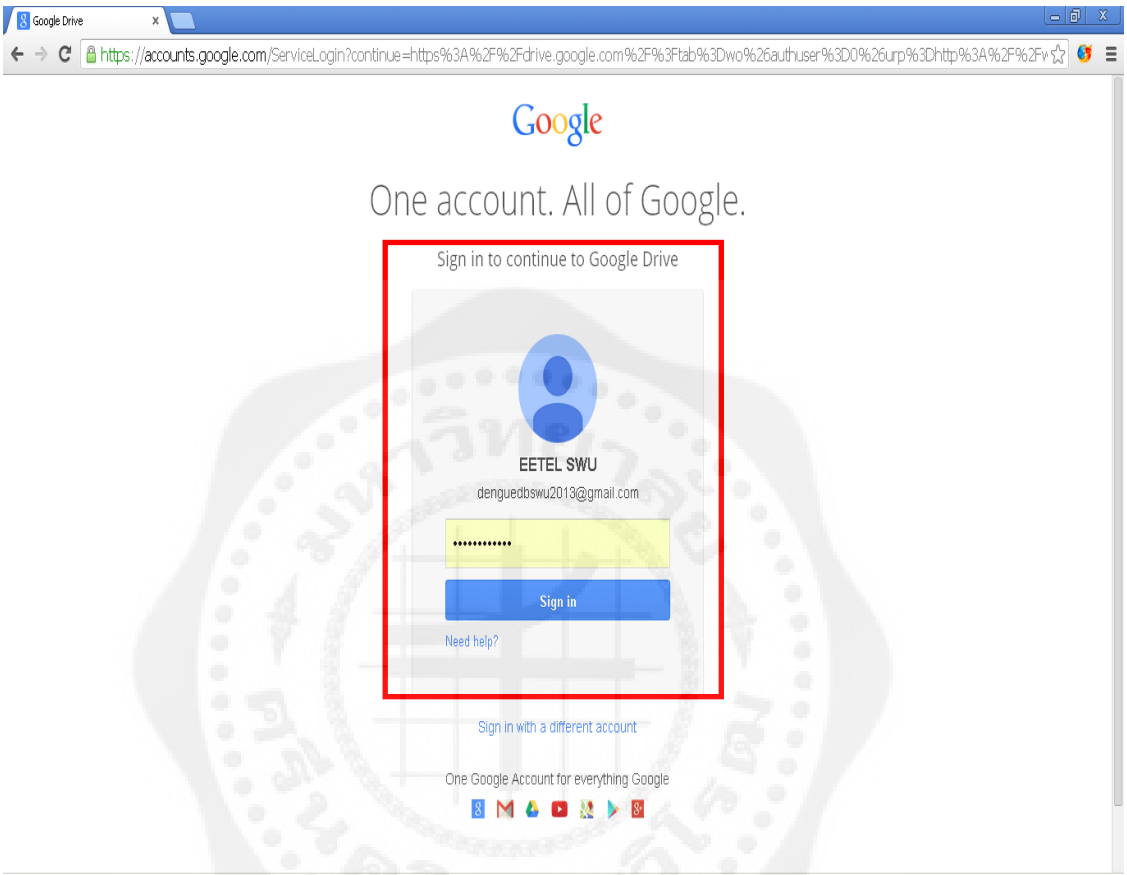


รูปที่ 2.13 เพิ่มรูปภาพให้กับโปรไฟล์ของ Gmail

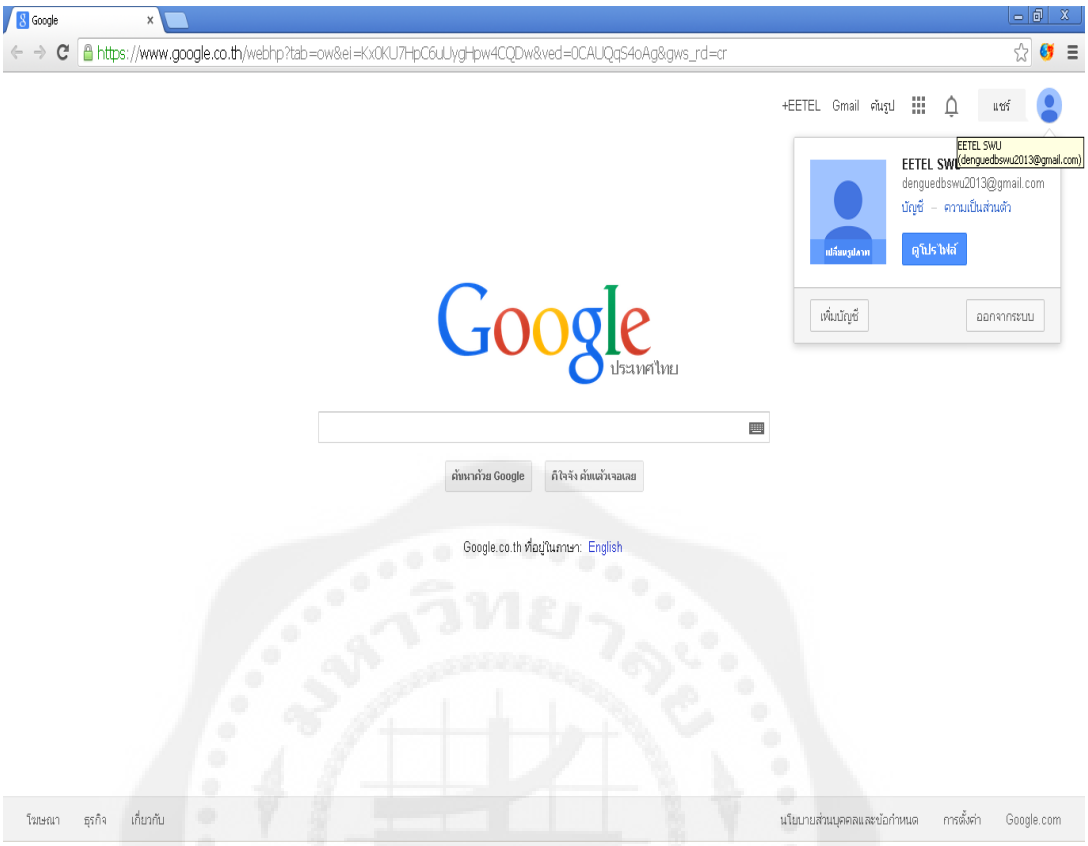
2.1.5 เมื่อเรียบร้อยแล้ว เข้ามาในส่วนสุดท้ายจะมีปุ่มให้เรายืนยันเพื่อเริ่มต้นการใช้งาน Gmail เป็นอันเสร็จเรียบร้อยสำหรับการสมัคร Gmail ดังรูปที่ 2.14

รูปที่ 2.14 ยืนยันเพื่อเริ่มต้นการใช้งาน Gmail

2.1.6 ล็อกอินเข้า Google โดย Google account พิมพ์ชื่อและรหัสลงไปแล้วกด Sign in ดังรูปที่ 2.15 และรูปที่ 2.16



รูปที่ 2.15 ล็อคอินเข้าใช้งาน Google apps



รูปที่ 2.16 ล็อคอิน Google account สำเร็จ

3. Google drive คือ การเก็บข้อมูลในรูปแบบของ Cloud storage หรือบริการจัดเก็บไฟล์ข้อมูลไว้บนเซิร์ฟเวอร์แล้วผู้ใช้งานสามารถเปิดใช้งานไฟล์ข้อมูลเหล่านั้นได้จากทุกที่ทุกเวลา

3.1 ข้อดีของ Google drive

3.1.1 ความสามารถในการแชร์ไฟล์ข้อมูลไปยังคนอื่น ๆ ได้ก่อนหน้านี้หากต้องการส่งไฟล์ประเภทต่าง ๆ ผ่านอีเมล์อาจต้องส่งหลายครั้งและเกิดความยุ่งยากในการจัดการ แต่ถ้าหากใช้ Google drive ปัญหานี้จะหมดไปเพราะไฟล์ข้อมูลจะถูกจัดเก็บไว้ที่เดียวบน Google drive และสามารถแชร์ไฟล์ให้คนอื่น ๆ โดยไม่ต้องส่งไฟล์กลับไปกลับมาให้เกิดความยุ่งยาก วิธีการนี้จะช่วยลดความซ้ำซ้อนในการส่งข้อมูลลงไปได้

3.1.2 ความสามารถในทำงานได้จากหลาย ๆ สถานที่ในเวลาที่พร้อม ๆ กัน เช่น นักข่าวในอเมริกากับนักข่าวที่ประเทศไทยสามารถทำงานบนไฟล์เดียวกันบน Google drive ได้และสามารถคุยโต้ตอบผ่านระบบ Chat เพื่อแลกเปลี่ยนความคิดเห็นเกี่ยวกับงาน

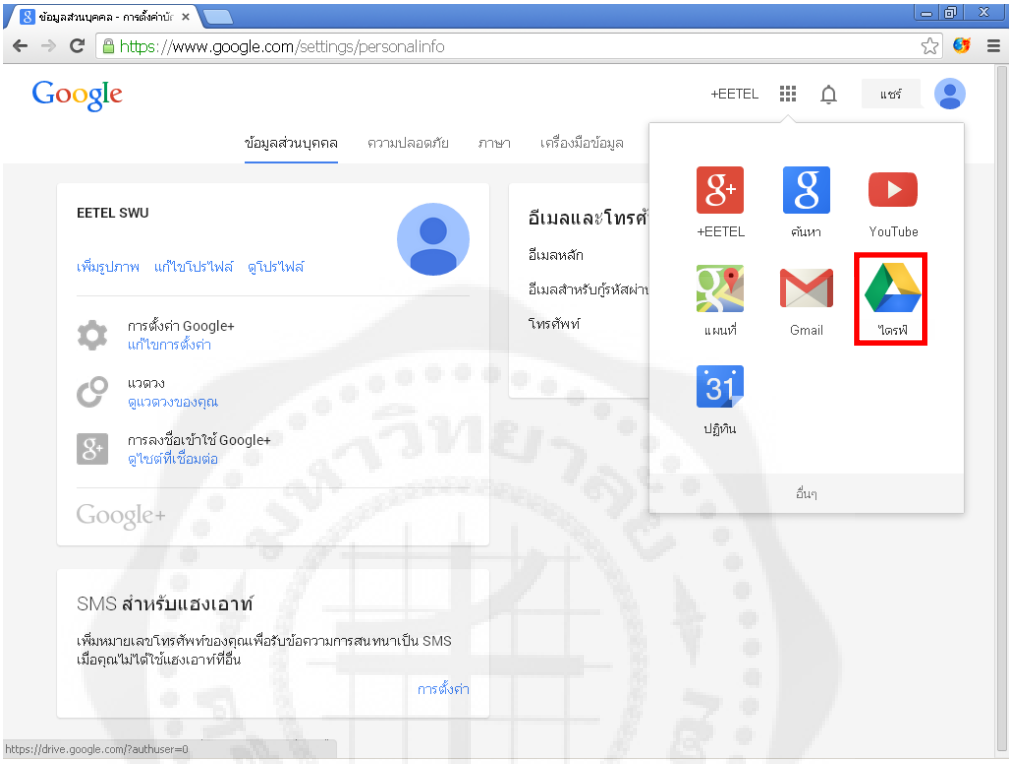
3.1.3 เป็นพื้นที่เก็บไฟล์ข้อมูลที่ปลอดภัยเนื่องจากไฟล์ข้อมูลถูกเก็บไว้บนเซิร์ฟเวอร์ของ Google หากคอมพิวเตอร์ที่บ้านของผู้ใช้งานได้รับความเสียหายไฟล์ข้อมูลเหล่านั้นก็จะไม่ได้รับความเสียหายตามไปด้วย

3.1.4 ระบบจัดเก็บไฟล์ข้อมูลของ Google drive จะแสดงรายละเอียดปริมาณข้อมูลแยกออกเป็น 3 ส่วนด้วยกันคือ Google drive,Gmail และ Google+ Photos ซึ่งมีพื้นที่รวมทั้งหมด 15 GB ผู้ใช้งานสามารถเข้าไปตรวจสอบปริมาณข้อมูลที่ใช้ไปจาก www.google.com/settings/storage ซึ่งหากใช้พื้นที่ไปจนครบจำนวน 15 GB ก็สามารถเพิ่มขนาดพื้นที่จัดเก็บข้อมูลได้ตามแพ็คเกจที่เหมาะสม

3.1.5 เมื่ออยู่ในสถานที่ที่ไม่สามารถเชื่อมต่ออินเตอร์เน็ตเช่น บนเครื่องบิน Google drive สามารถเปิดไฟล์ข้อมูลขึ้นมาทำงานแบบออฟไลน์ได้ เมื่อลงจากเครื่องและสามารถเชื่อมต่ออินเตอร์เน็ตได้ก็สามารถอัพเดทไฟล์งานนั้นขึ้นสู่ Google drive ได้

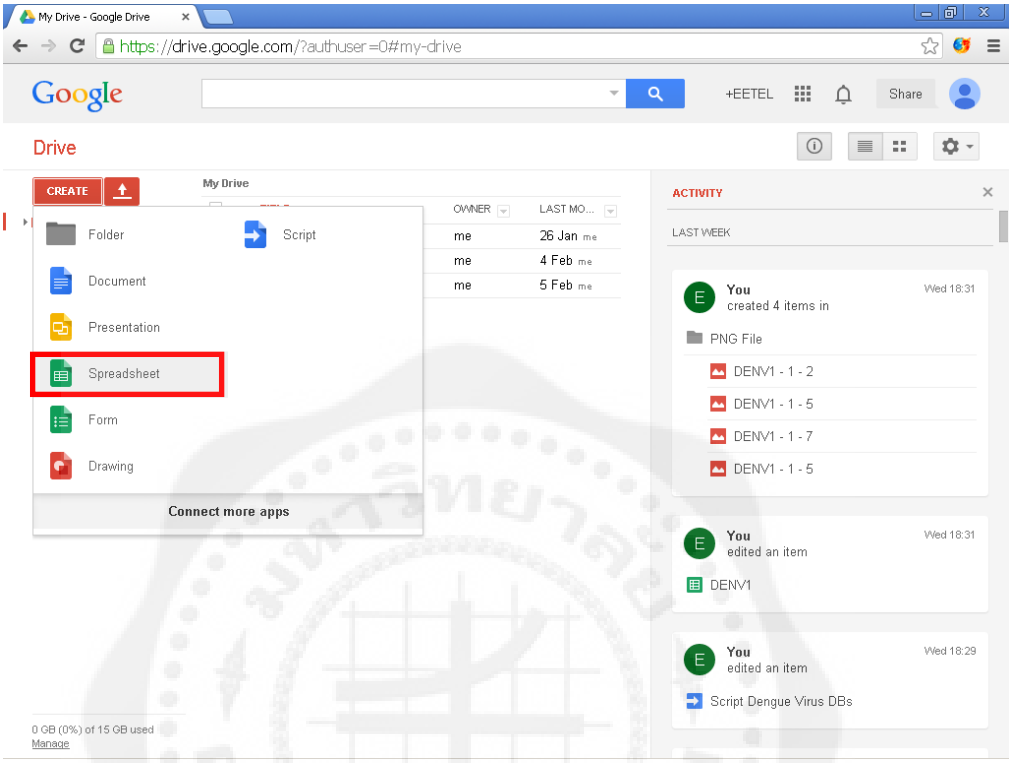
3.2 การใช้งาน Google drive ในการเก็บข้อมูล

3.2.1 Import ข้อมูลลงใน Spreadsheet ให้เข้าไปที่ Google drive ดังรูปที่ 2.17



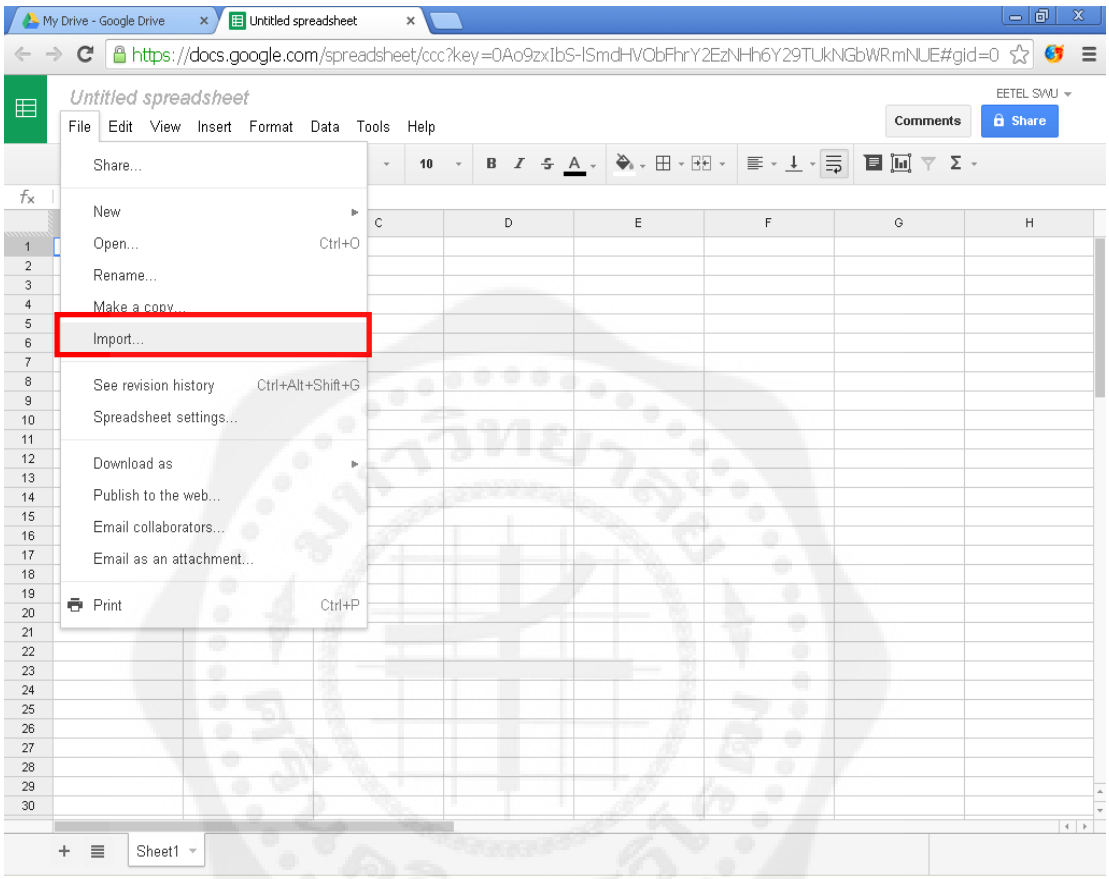
รูปที่ 2.17 การเข้า Google drive ด้วย Gmail

3.2.2 ทำการสร้าง Spreadsheet ดังรูปที่ 2.18



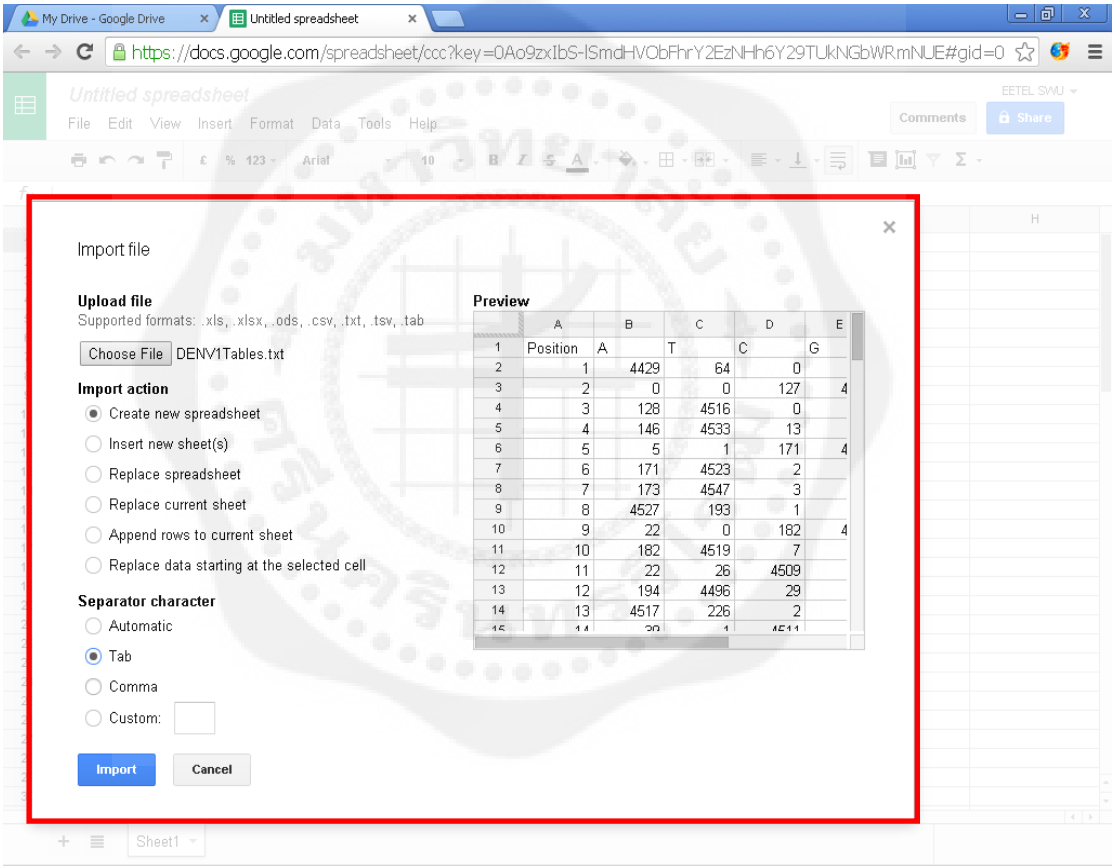
รูปที่ 2.18 การสร้าง Spreadsheet บน Google drive

3.2.3 จะเป็นเว็บเพจที่เหมือนกับหน้าตาโปรแกรม Microsoft excel ให้เราคลิกที่ File และ คลิกที่ Import… ดังรูปที่ 2.19



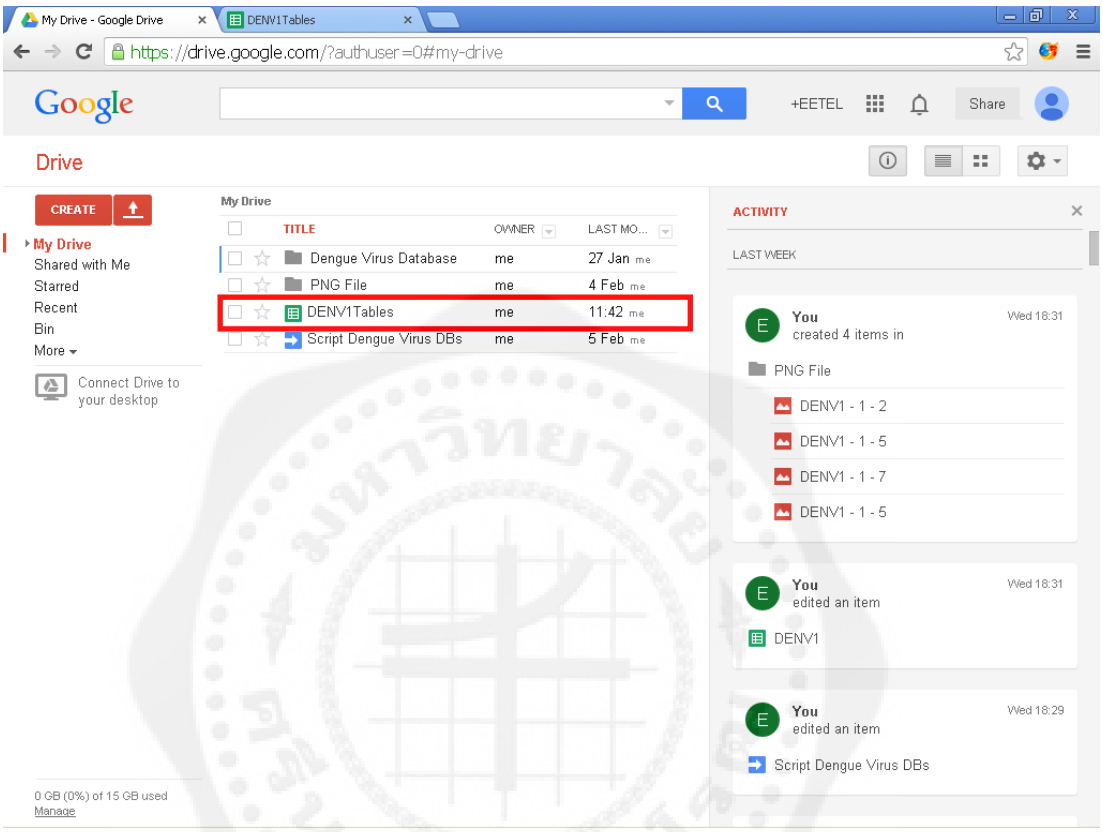
รูปที่ 2.19 Google spreadsheet

3.2.4 คลิก Choose file เลือกไฟล์ที่เราต้องการอัพโหลดไฟล์ข้อมูลลงใน Google spreadsheet ขึ้นมา ซึ่งจะรับไฟล์ประเภท .xls .xlsx .ods .csv .txt .tsv .tab ตัวอย่างเช่น ถ้าเราต้องการ Import ไฟล์ Tab-delimited import ชื่อ DENV1Tables.txt ได้โดยในส่วน Upload file คลิกที่ Choose file DENV1Tables.txt แล้ว Open ขึ้นมาจะได้ภาพตัวอย่างไฟล์ (Preview) ขึ้นอยู่ทางขวาของหน้าต่าง ส่วนของ Import action ให้เลือกแบบ Create new spreadsheet และส่วนของ Separator character เลือกแบบ Tab แล้วคลิก Import ดังรูปที่ 2.20

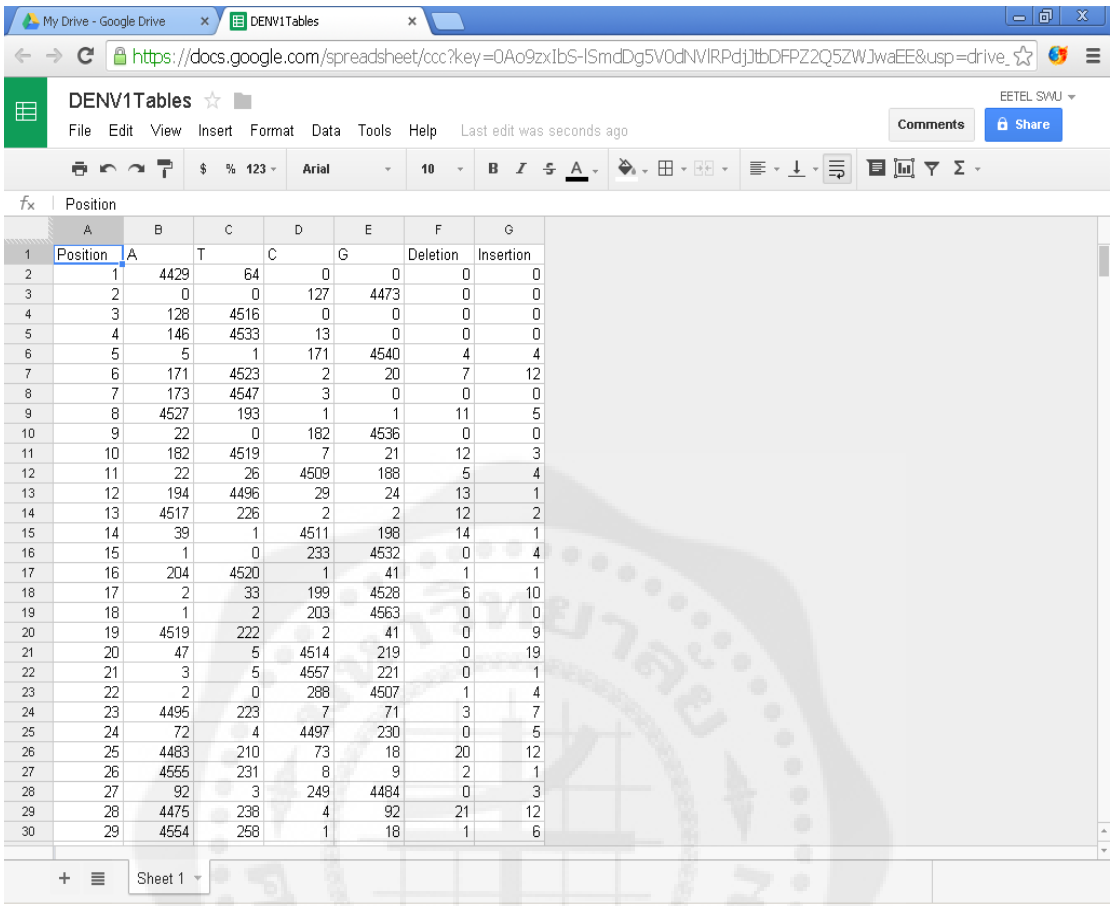


รูปที่ 2.20 การ Import file ลงบน Google spreadsheet

3.2.5 จะได้ข้อมูลที่ต้องการไว้ใน Google spreadsheet อยู่บน Google drive ดังรูปที่ 2.21 และรูปที่ 2.22



รูปที่ 2.21 ไฟล์ที่ถูก Import ลงบน Google drive



	A	B	C	D	E	F	G
1	Position	A	T	C	G	Deletion	Insertion
2	1	4429	64	0	0	0	0
3	2	0	0	127	4473	0	0
4	3	128	4516	0	0	0	0
5	4	146	4533	13	0	0	0
6	5	5	1	171	4540	4	4
7	6	171	4523	2	20	7	12
8	7	173	4547	3	0	0	0
9	8	4527	193	1	1	11	5
10	9	22	0	182	4536	0	0
11	10	182	4519	7	21	12	3
12	11	22	26	4509	188	5	4
13	12	194	4496	29	24	13	1
14	13	4517	226	2	2	12	2
15	14	39	1	4511	198	14	1
16	15	1	0	233	4532	0	4
17	16	204	4520	1	41	1	1
18	17	2	33	199	4528	6	10
19	18	1	2	203	4563	0	0
20	19	4519	222	2	41	0	9
21	20	47	5	4514	219	0	19
22	21	3	5	4557	221	0	1
23	22	2	0	288	4507	1	4
24	23	4495	223	7	71	3	7
25	24	72	4	4497	230	0	5
26	25	4483	210	73	18	20	12
27	26	4555	231	8	9	2	1
28	27	92	3	249	4484	0	3
29	28	4475	238	4	92	21	12
30	29	4554	258	1	18	1	6

รูปที่ 2.22 ข้อมูลใน Google spreadsheet ที่ได้จากการ Import

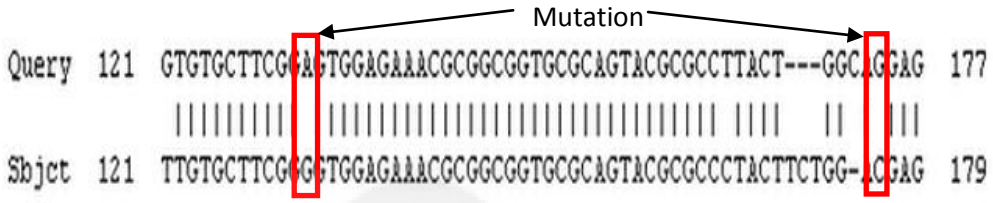
4. MicroRNA (miRNA)

เชื้อไวรัสไข้เลือดออกจะมีลักษณะทางพันธุกรรมที่เรียกว่า miRNA ซึ่งประกอบด้วยหน่วยพันธุกรรมคือ A T C G มาเรียงต่อๆกันเป็นสายหรือเป็นเส้นๆซึ่งมีความยาวกว่าหมื่นหน่วยในการถอดรหัสนั้นจำเป็นต้องอาศัยเทคนิคทางเคมีหรือชีววิทยาซึ่งพบว่ามีอุปสรรคคือ เทคโนโลยีในปัจจุบันนั้นสามารถดึงรหัสทางพันธุกรรมนั้นออกมาได้เพียงแค่ท่อนสั้นๆเหมือนจิ๊กซอแล้วนำมาต่อกันได้ ดังรูปที่ 2.23

รูปที่ 2.23 สายรหัสทางพันธุกรรม miRNA

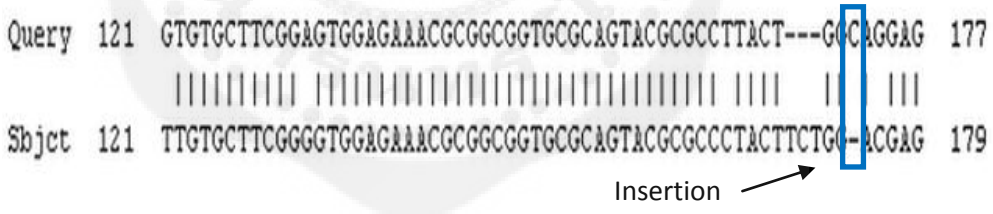
อีกทั้งลักษณะทางพันธุกรรมโดยทั่วไปอาจเกิดการกลายพันธุ์ได้ โดยการกลายพันธุ์ก็คือการที่รหัสพันธุกรรม miRNA มีหน่วยพันธุกรรมบางหน่วยเปลี่ยนแปลงไปสามารถเกิดขึ้นได้ 3 รูปแบบ ได้แก่

4.1 มิวเทชัน (Mutation) คือ การที่รหัสทางพันธุกรรมตัวใดตัวหนึ่งเกิดเปลี่ยนแปลงไปเป็นตัวอื่นซึ่งก็หมายความว่า เช่น จากรหัสพันธุกรรม ATTCGGCA เปลี่ยนแปลงเป็น ATGCGGTA จากรหัสพันธุกรรมที่เกิดขึ้นใหม่พบว่า ตำแหน่งที่ 3 เปลี่ยนจาก T เป็น G และตำแหน่งที่ 7 เปลี่ยนจาก C เป็น T หรือสามารถสังเกตการณ์เปลี่ยนแปลงได้ดังรูป



รูปที่ 2.24 การเกิดมิวเทชัน

4.2 อินเซริสชัน (Insertion) คือ เมื่อรหัสพันธุกรรมมีการเพิ่มขึ้นจากเดิม ตัวอย่างเช่น จากรหัสพันธุกรรม ATTCGGCA เปลี่ยนเป็น ATTCGGCTA จากรหัสพันธุกรรมที่เกิดขึ้นใหม่พบว่าตำแหน่งที่แปดมี T เกินมาหนึ่งหน่วย หรือสามารถสังเกตการเปลี่ยนแปลงได้ ดังรูป

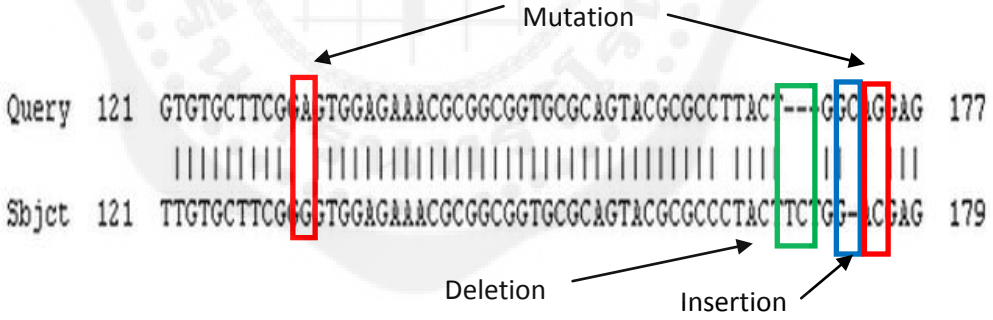


รูปที่ 2.25 การเกิดอินเซริสชัน

4.3 ดีลีทชัน (Deletion) คือ เมื่อรหัสพันธุกรรมมีหน่วยใดๆสูญหายไปจากเดิม ตัวอย่างเช่น จากรหัสพันธุกรรม ATTCGGCA เปลี่ยนเป็น ATCGGCA จากรหัสพันธุกรรมที่เกิดขึ้นใหม่พบว่าที่ตำแหน่งที่สองหรือสาม T หายไปหนึ่งหน่วย หรือสามารถสังเกตการเปลี่ยนแปลงได้จากรูปด้านล่าง

รูปที่ 2.26 การเกิดดีลีทชัน

ซึ่งสายรหัสพันธุกรรมเส้นเดียวกันจะสามารถเปลี่ยนแปลงได้ทั้งสามประเภทที่กล่าวมาคือ ในรหัสพันธุกรรมหนึ่งสายนั้นสามารถเกิด มิวเทชัน อินเซริสชัน และดีลีทชัน ขึ้นในสายเดียวกันก็ได้ ดังรูป



รูปที่ 2.27 การเปลี่ยนแปลงของรหัสทางพันธุกรรม

5. Model View Controller (MVC)

Model View Controller (MVC) จะแบ่งแอพพลิเคชันตามบทบาทหน้าที่ (Roles of Objects) โดยแบ่งออกเป็น 3 บทบาทด้วยกัน คือ

5.1 Model (M) คือ ออบเจกต์ที่ทำหน้าที่เป็นตัวแทนข้อมูล ไม่ว่าข้อมูลจะถูกจัดเก็บในรูปแบบใดในระบบฐานข้อมูลหรือในไฟล์เมื่อข้อมูลนั้นถูกโหลดเข้ามาในแอพพลิเคชั่น เราจะเปลี่ยนมันให้อยู่ในรูปของออบเจกต์ และเราเรียกบทบาทของออบเจกต์นี้ว่า Model ยกตัวอย่างเช่น ออบเจกต์ Customer Employee Product เป็นต้น

5.2 View (V) คือ ออบเจกต์ที่ทำหน้าที่ในการแสดงผล เช่น แสดงหน้าฟอร์มกรอกข้อมูล แสดงผลลัพธ์ของการค้นหา เป็นต้น หรือ View ก็คือ User Interface (UI) นั่นเอง

5.3 Controller (C) คือ ออบเจกต์ที่ทำหน้าที่รับคำสั่ง เช่น คำสั่งค้นหา คำสั่งลบข้อมูล เป็นต้น เมื่อมีคำสั่งเข้ามาจากผู้ใช้ออบเจกต์ตัวนี้จะทำหน้าที่รับคำสั่งและเรียกใช้ออบเจกต์ตัวอื่นๆให้ทำงานร่วมกัน

6. Django

เป็นแพลตฟอร์มที่มีความเป็นระเบียบในการเขียนโปรแกรม และง่ายต่อการแก้ไขโปรแกรมในภายหลังมีแบบแผนที่ทำให้สร้างเว็บแอพพลิเคชันให้สำเร็จได้เร็ว ใช้ความสามารถจากภาษาไพธอนในการเขียนโปรแกรม เชื่อมต่อกับฐานข้อมูลที่สามารถรองรับจำนวนข้อมูลที่เพิ่มมากขึ้นได้

6.1 วิธีการใช้งาน Django

6.1.1 การสร้างโปรเจค (Creating a project)

ใน Command line เข้าไปในไดเรคทอรี่ที่เก็บโค้ดไว้แล้วรัน Command ดังนี้

```
django-admin.py startproject mysite
```

เมื่อรัน startproject สำเร็จแล้วจะพบไฟล์ดังนี้

```
mysite/
    manage.py
    mysite/
        __init__.py
        settings.py
```

urls.py

wsgi.py

mysite/ คือ รูทไดเรคทอรี่คือไดเรคทอรี่ที่เก็บโปรเจคไว้สามารถเปลี่ยนเป็นชื่อตามต้องการได้manage.py คือ command-line เอาไว้รันโปรแกรม Django

mysite/ __init__.py คือ ไฟล์เปล่าที่เอาไว้เรียก Python ไดเรกทอรี่และพิจารณาชื่อแพคเกจของPython

mysite/ setting.py คือ ไฟล์ที่เก็บค่าคอนฟิคของโปรแกรม Django ไว้สามารถเข้าไปแก้ไขปรับปรุงตามการใช้งานได้

mysite/ wsgi.py คือ ไฟล์ที่เอาไว้ Deploy django ขึ้นไปบน Server

6.1.2 การรันโปรแกรม (Run server)

ถ้าไม่มีอะไรผิดพลาด และถ้าทำทุกอย่างเสร็จสมบูรณ์แล้วจะรันโปรแกรมให้พิมพ์ Python manage.py runserver ใน Command-line จะพบข้อความดังนี้

Validating models...

0 errors found

March 20, 2014 - 15:50:53

Django version 1.6, using settings 'mysite.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CONTROL-C.

หลังจากนั้นเปิด Web browser พิมพ์ http://127.0.0.1:8000/ ลงใน URL และจะพบข้อความ “Welcome to Django” บนหน้าเว็บเพจแสดงว่าโปรแกรมไม่ผิดพลาด

6.1.3 ตั้งค่าฐานข้อมูล (Database setup)

แก้ไข setting.py ซึ่งภายในไฟล์โดยปกติก็คือ Python Module แต่ถูกทำให้อยู่ในรูปของ Django setting โดยค่าปกติจะคอนฟิคไปที่ SQLite ซึ่งจะไม่ต้องติดตั้งโปรแกรม Database ใดๆเลยเนื่องจาก SQLite มีอยู่ใน Python อยู่แล้ว แต่ถ้าไม่ใช้ SQLite ก็จะต้องใส่ค่า USER PASSWORD HOST และอีกมากมาย หลังจากใส่ค่าลงไปในไฟล์ setting.py หมดแล้ว รัน python manage.py syncdb บน Command-line เพื่อเป็นการซิงค์ข้องมูลทั้งหมดไปยัง Database ที่ใช้

6.1.4 สร้างโมเดล (Creating model)

เข้าไปในไดเรคทอรี่ที่มีไฟล์ manage.py อยู่และรัน Command ต่อไปนี้บน Command-line

python manage.py startapp polls และจะพบโฟลเดอร์ภายในประกอบด้วยไฟล์ต่อไปนี้

polls/

__init__.py

admin.py

models.py

tests.py

views.py

6.1.4.1 ไฟล์ models.py เป็นไฟล์ที่เรียกว่า โมเดล ตามรูปแบบ MVC ของ Django คือตัว Model เป็นไฟล์ที่ใช้สำหรับสร้างฟิลด์หรือจัดการกับฐานข้อมูลใช้ออกแบบฐานข้อมูลผ่านทางไฟล์นี้

6.1.4.2 ไฟล์ views.py เป็นไฟล์ที่เรียกว่า วิว ตามรูปแบบ MVC ของ Django คือตัว View เป็นไฟล์ที่ใช้สำหรับสร้างหรือกำหนดฟังก์ชันต่างๆที่ใช้ในการแสดงผลออกมาทางหน้าจอ

6.2 สร้างเทมเพลต (Creating template)

สร้างโฟลเดอร์ชื่อว่า templates ไว้ในโฟลเดอร์ polls และสร้างโฟลเดอร์ polls ไว้ในโฟลเดอร์templates อีกที ซึ่งภายในไฟล์ polls จะเป็นไดเรคทอรี่ที่เอาไว้เก็บไฟล์ HTML CSS หรือไฟล์ที่เกี่ยวข้องกับการสร้างหน้าเว็บไซต์ จากนั้นเข้าไปในไฟล์ setting.py ใส่ไดเรคทอรี่ของ templates ลงใน TEMPLATE_DIRS

6.3 สร้าง URL

เข้าไปในไฟล์ mysite/urls.py พิมพ์คำสั่งดังนี้ลงไปในส่วนของ urlpatterns =

patterns(‘’,

url(r'^polls/', include('polls.urls', namespace="polls")),

url(r'^admin/', include(admin.site.urls)),

)

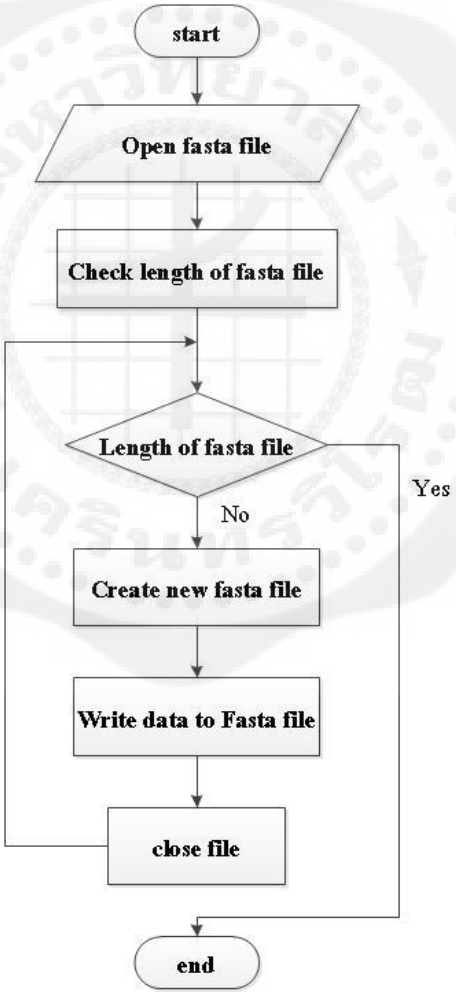
6.4 รันเซิร์ฟเวอร์ (Running server)

พิมพ์ python manage.py startserver บน command-line และเปิด Web browser พิมพ์ http://127.0.0.1/polls ลง ในช่อง URL จะพบหน้าเว็บที่สร้างจากไฟล์ HTML ไว้

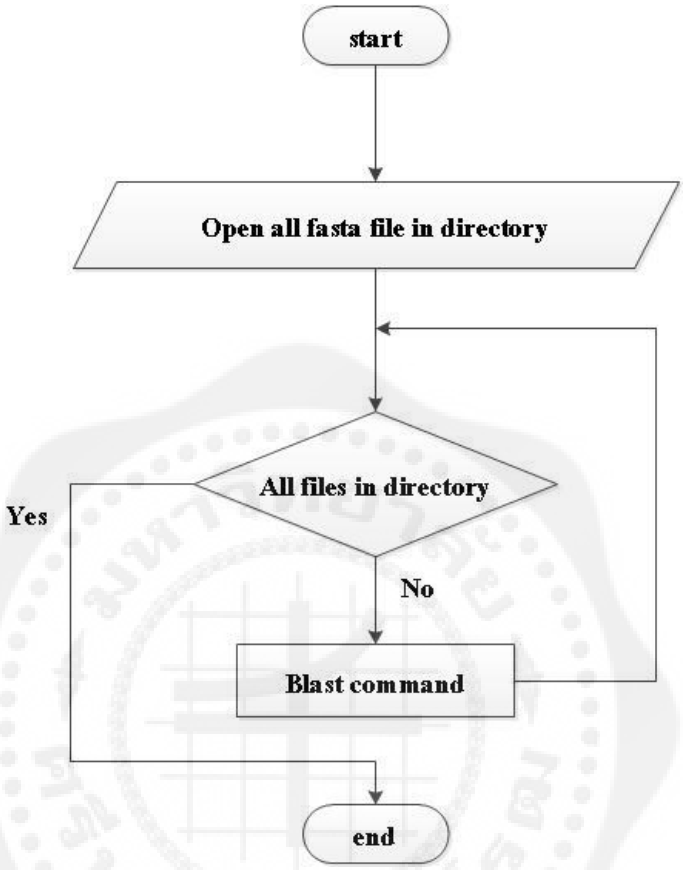
บทที่ 3
ขั้นตอนการสร้างและวิธีการสร้าง

1. ขั้นตอนการจัดการไฟล์ฐานข้อมูลด้วยภาษาไพธอน

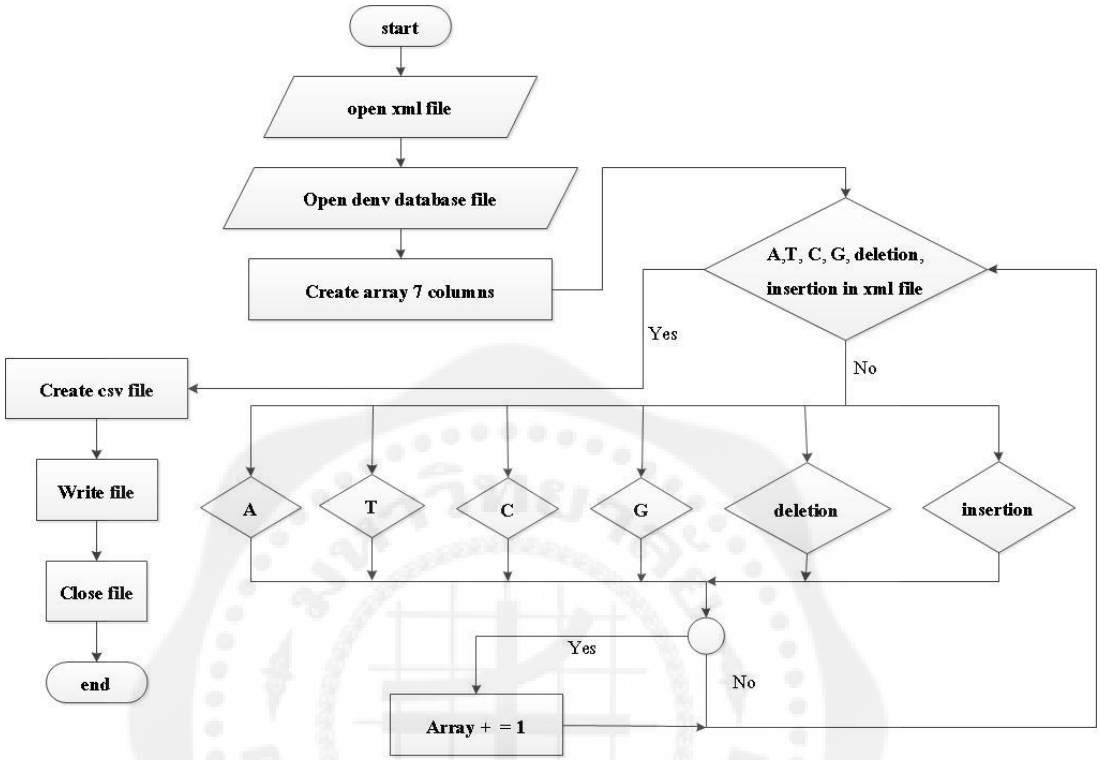
1.1 เริ่มจากเขียนโปรแกรมแยกไฟล์ข้อมูล (.fna) ที่ภายในไฟล์มีข้อมูลของกลุ่มผู้ติดเชื้อไวรัสไข้เลือดออก ออกเป็นไฟล์ (.fasta) หนึ่งไฟล์ต่อผู้ป่วยหนึ่งคน



1.2 นำไฟล์ Fasta ที่ได้มาทำให้เป็นไฟล์ XML

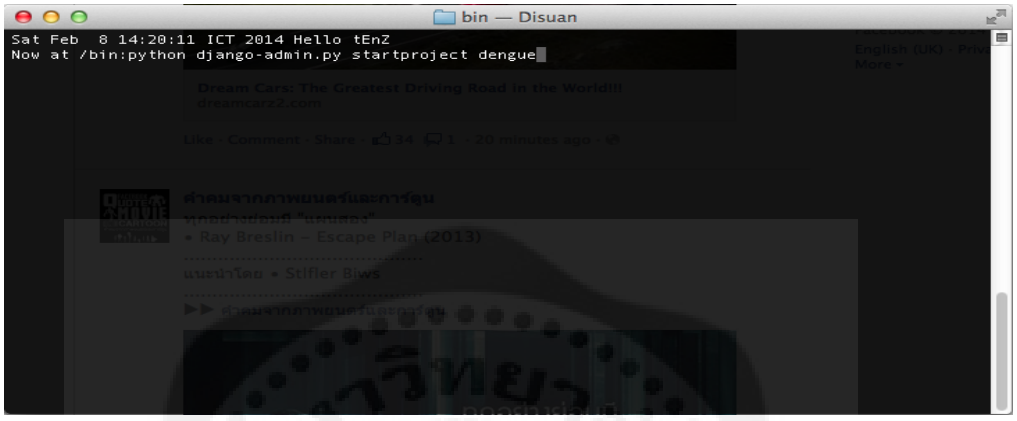


1.3 นำไฟล์ XML มาบันทึกข้อมูลผู้ป่วยลงในตารางเป็นไฟล์ CSV



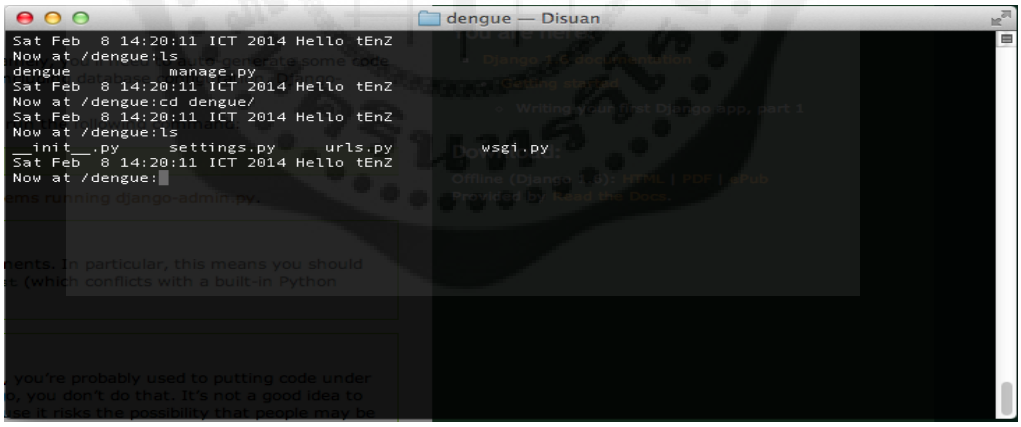
2. วิธีการสร้างเว็บแอพพลิเคชันด้วยจังโก้ (Django)

2.1 เริ่มจากสร้างโปรเจคด้วยการรันคำสั่ง python django-admin.py startproject dengue บน Command line ดังรูปที่ 3.1



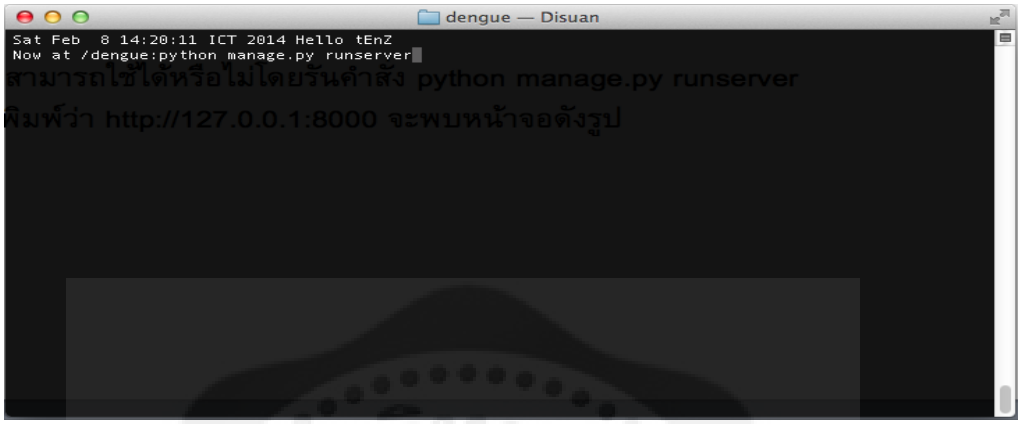
รูปที่ 3.1 คำสั่งสร้างโปรเจค

2.2 หลังจากสร้างโปรเจคขึ้นมาแล้วลองเข้าไปดูในโฟลเดอร์จะพบว่ามีไฟล์ ดังรูปที่ 3.2

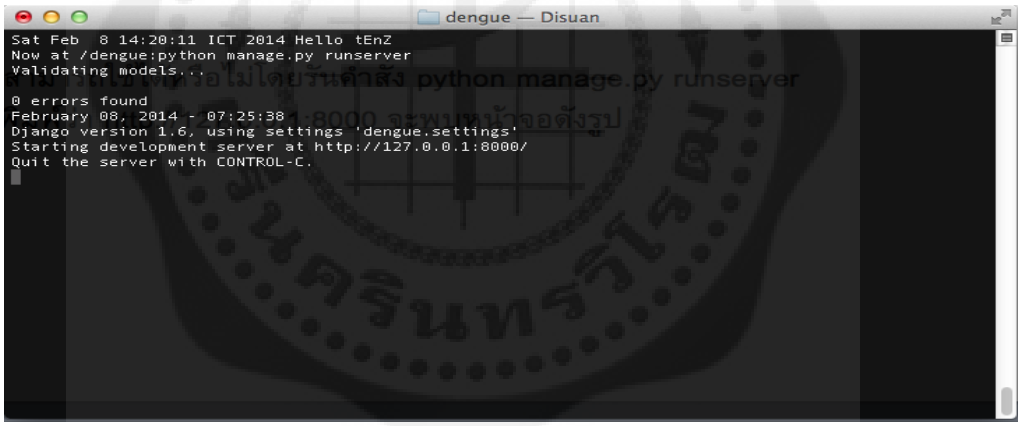


รูปที่ 3.2 ไฟล์ในโฟลเดอร์โปรเจค

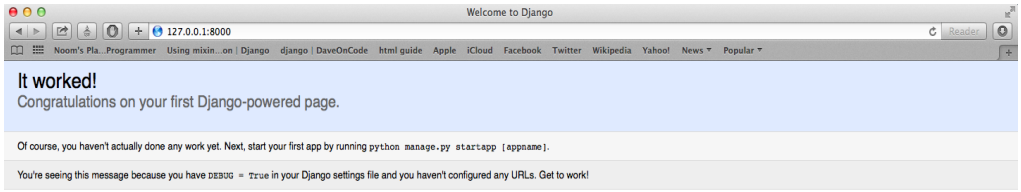
2.3 ตรวจสอบว่าโปรแกรมสามารถใช้ได้หรือไม่โดยรันคำสั่ง python manage.py runserver จากนั้นเปิด Browser แล้วพิมพ์ว่า http://127.0.0.1:8000 จะพบหน้าจอดังรูปที่ 3.3 – 3.5



รูปที่ 3.3 คำสั่งรันเซิร์ฟเวอร์

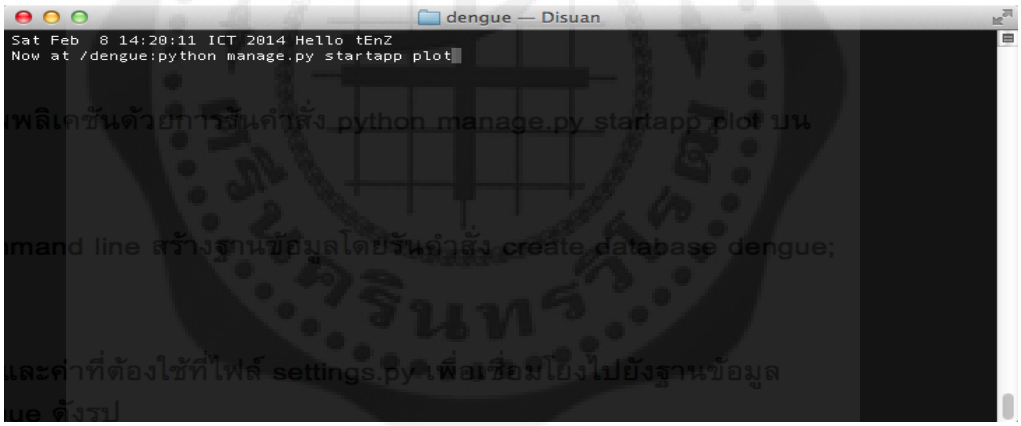


รูปที่ 3.4 รันคำสั่งรันเซิร์ฟเวอร์

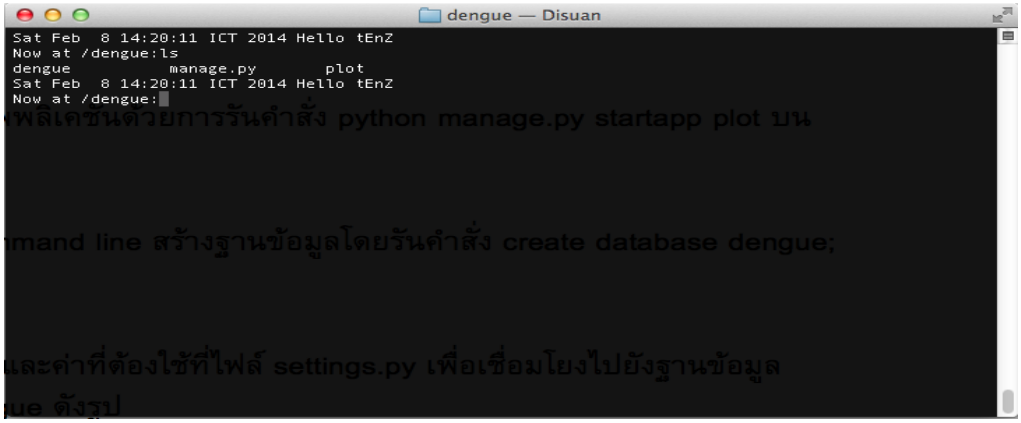


รูปที่ 3.5 เซิร์ฟเวอร์สามารถใช้งานได้

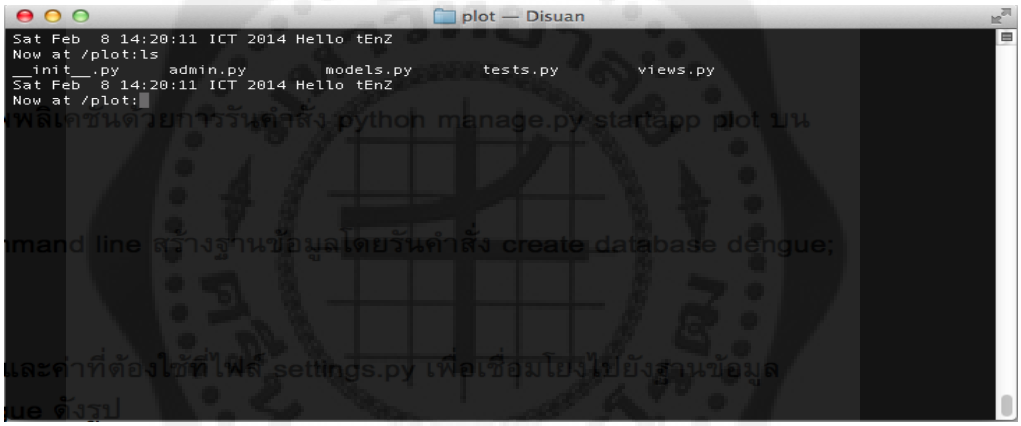
2.4 สร้างโฟลเดอร์เว็บแอพพลิเคชันด้วยการรันคำสั่ง python manage.py startapp plot บน Command line ดังรูปที่ 3.6 – 3.8



รูปที่ 3.6 คำสั่งสตาร์ทแอพ plot

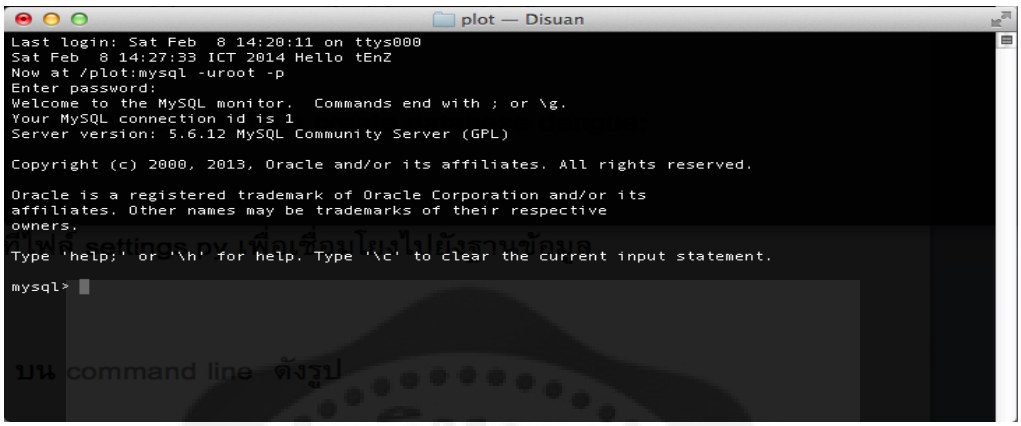


รูปที่ 3.7 โฟลเดอร์ plot

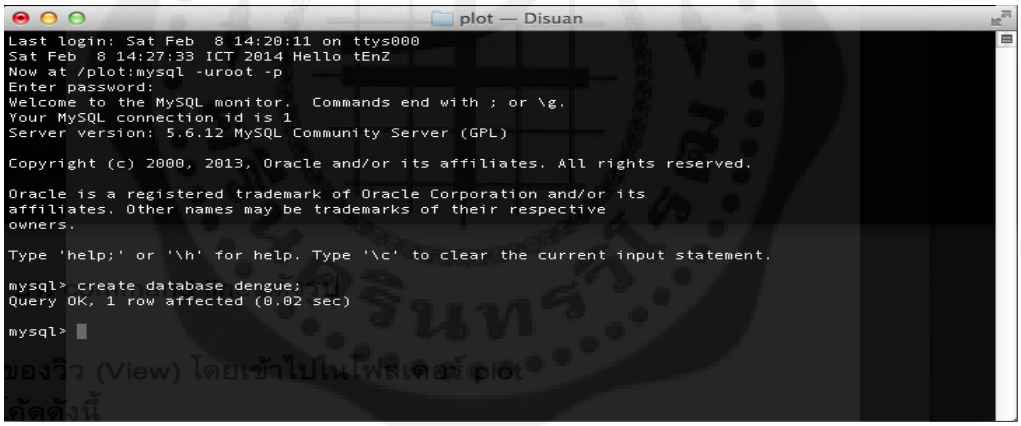


รูปที่ 3.8 ภายในโฟลเดอร์ plot

2.5 เข้า MySQL จาก command line สร้างฐานข้อมูลโดยรันคำสั่ง create database dengue; บน command line ดังรูปที่ 3.9 – 3.10

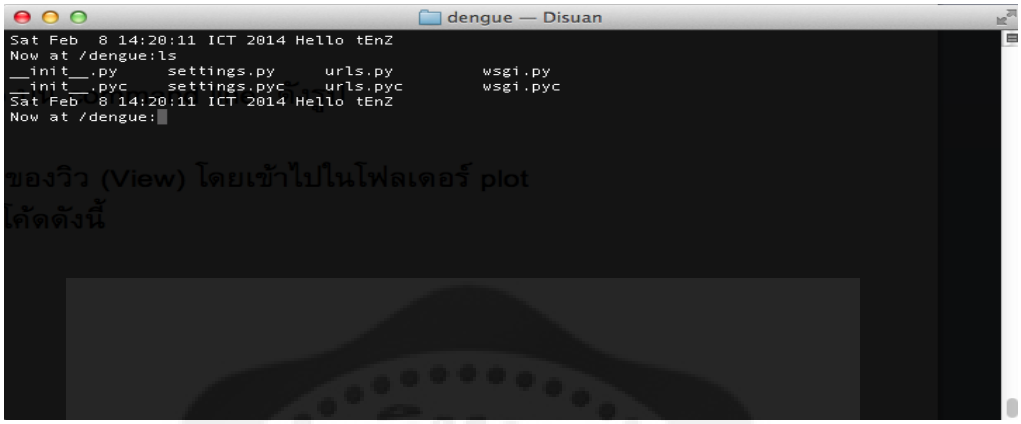


รูปที่ 3.9 เปิดโปรแกรม MySQL

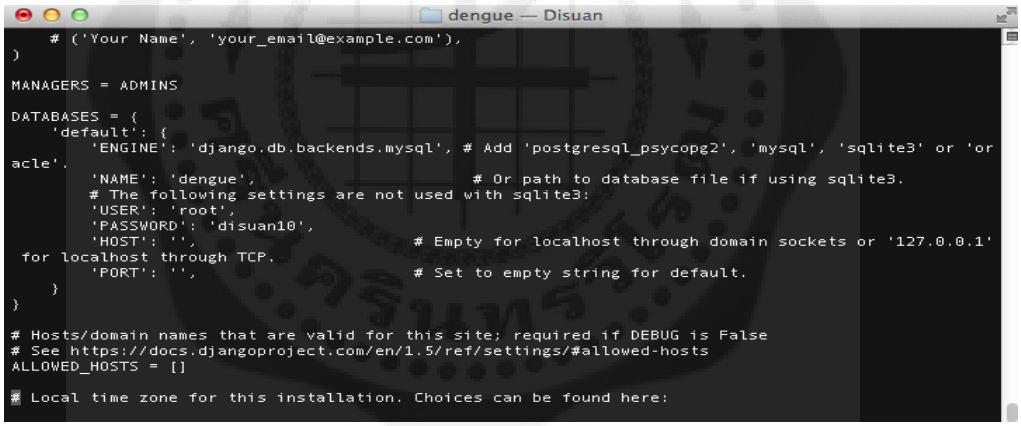


รูปที่ 3.10 รันคำสั่งสร้าง Database ใน MySQL

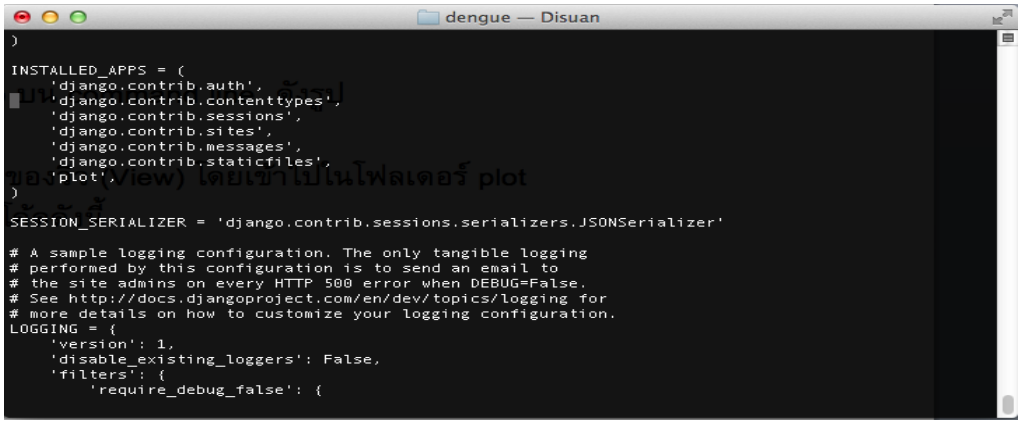
2.6 เข้าไปตั้งค่าฐานข้อมูลและค่าที่ต้องใช้ที่ไฟล์ settings.py เพื่อเชื่อมโยงไปยังฐานข้อมูล MySQL ในโฟลเดอร์ Dengue ดังรูปที่ 3.11 – 3.13



รูปที่ 3.11 ไฟล์ settings.py

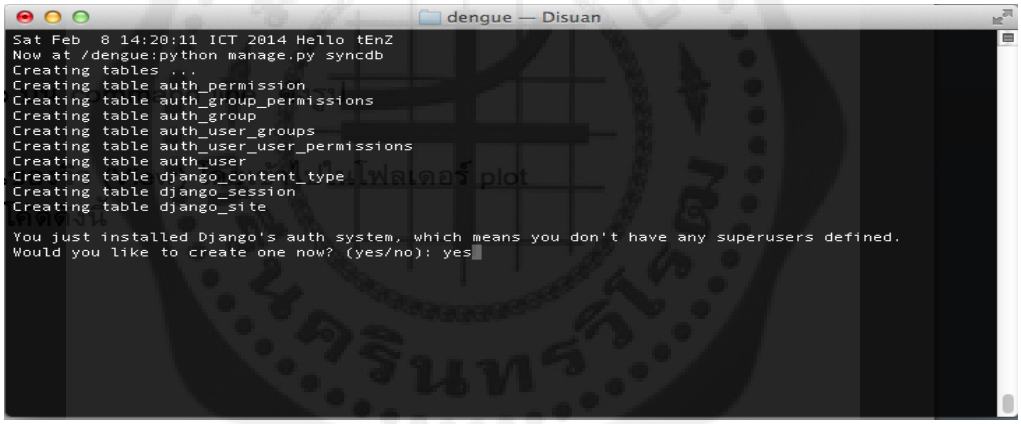


รูปที่ 3.12 ใส่ชื่อฐานข้อมูลและพาสเวิร์ด

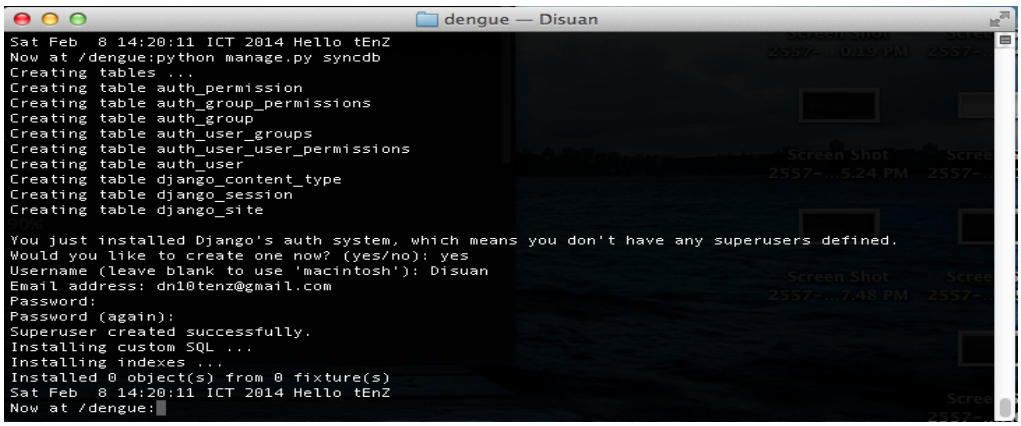


รูปที่ 3.13 ใส่ชื่อแอพ plot ที่สร้างขึ้น

2.7 รันคำส่ง python manage.py syncdb บน Command line ดังรูปที่ 3.14 – 3.15

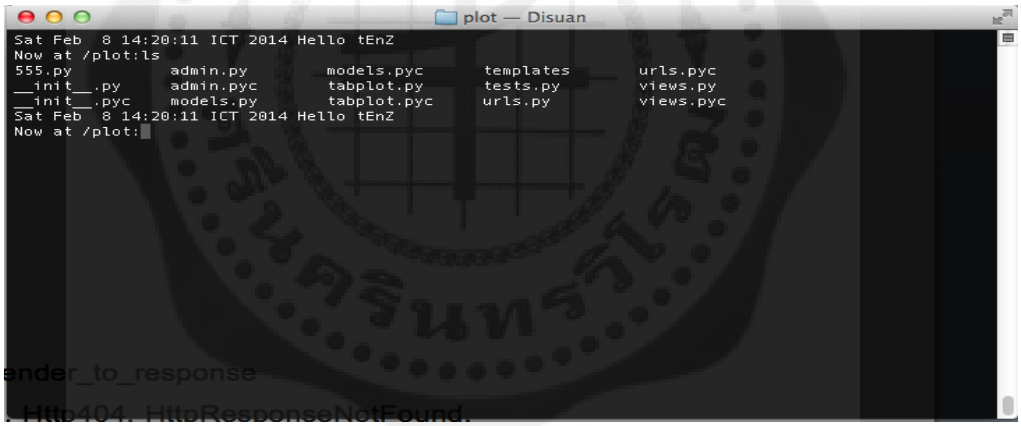


รูปที่ 3.14 คำสั่ง syncdb



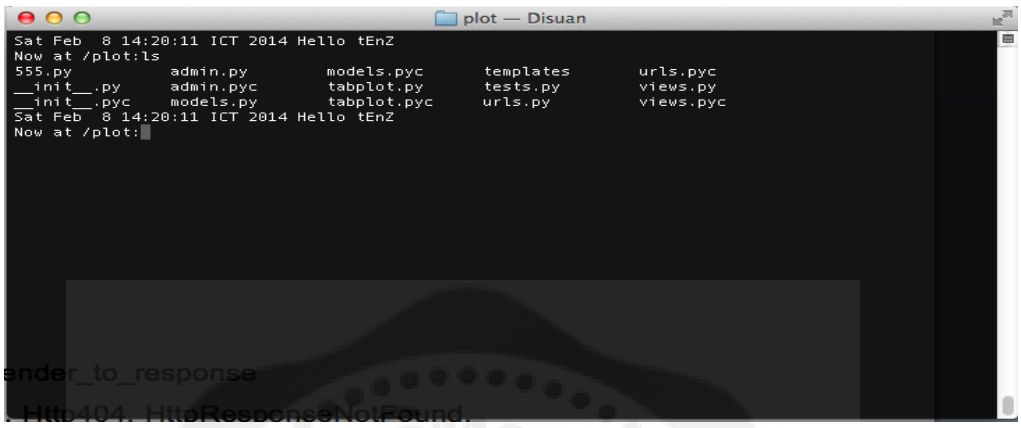
รูปที่ 3.15 รันคำสั่ง syncdb

2.8 ขั้นตอนต่อไปเป็นการเริ่มสร้างในส่วนของวิว โดยเข้าไปในโฟลเดอร์ plot จากนั้นสร้างไฟล์ tabplot.py ขึ้นมา ดังรูปที่ 3.16



รูปที่ 3.16 สร้างไฟล์ tabplot.py

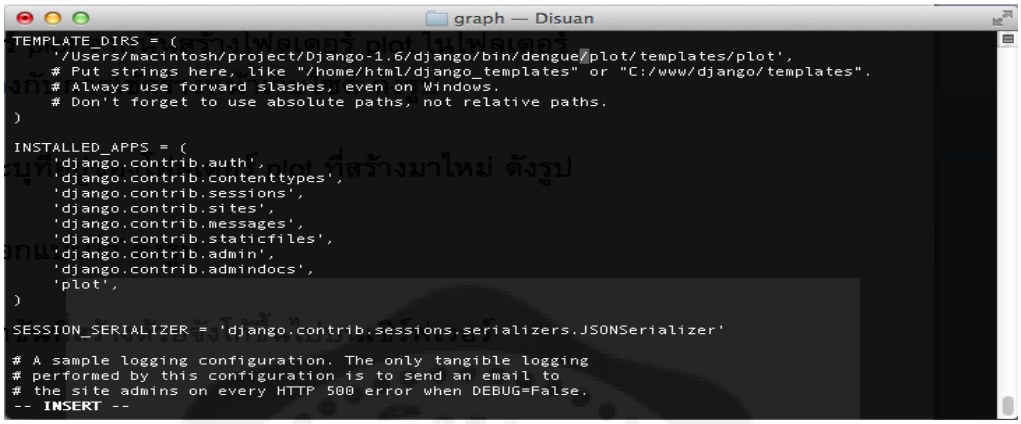
2.9 สร้างโฟลเดอร์ templates ในโฟลเดอร์ plot จากนั้นสร้างโฟลเดอร์ plot ในโฟลเดอร์ templates อีกที เพื่อใช้เก็บไฟล์ที่เกี่ยวข้องกับการใช้สร้างหน้าเว็บไซต์ ดังรูปที่ 3.17 – 3.18



รูปที่ 3.17 โฟลเดอร์ templates

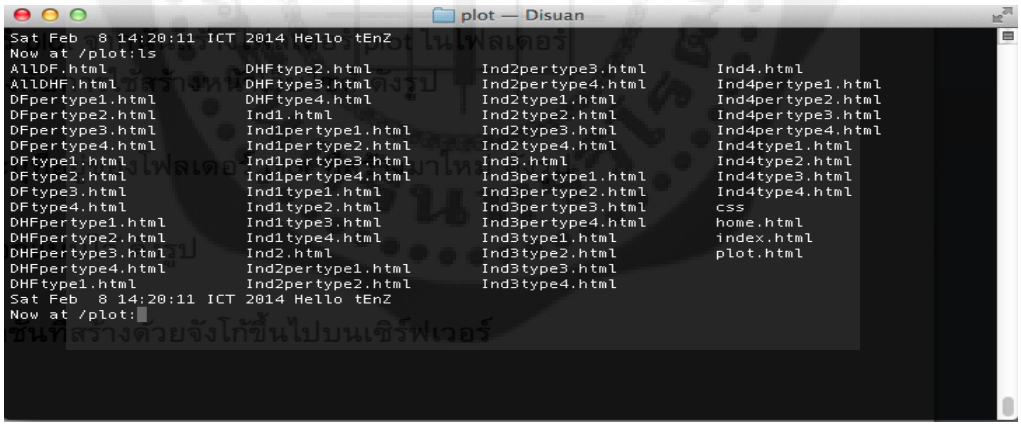
รูปที่ 3.18 โฟลเดอร์ plot

2.10 เปิดไฟล์ settings.py อีกครั้งตั้งค่าระบุที่อยู่ของโฟลเดอร์ plot ที่สร้างมาใหม่ ดังรูปที่ 3.19



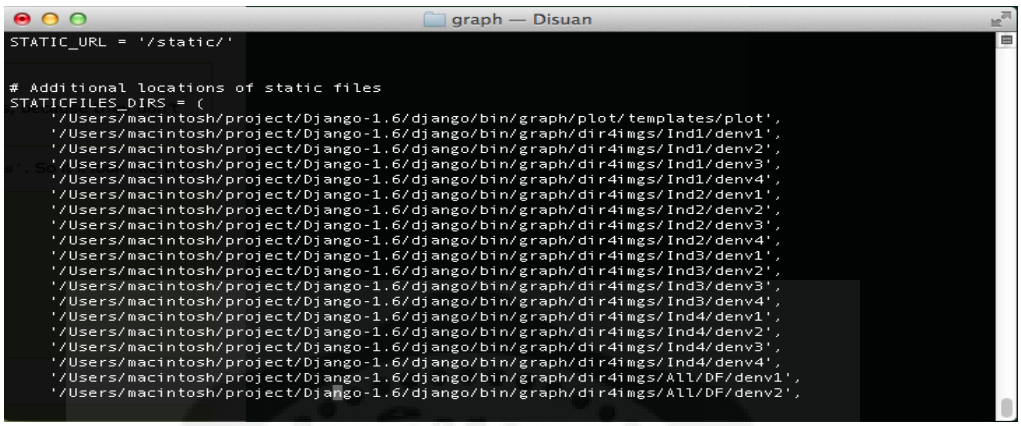
รูปที่ 3.19 ตั้งค่า TEMPLATE_DIRS

2.11 ทำการสร้าง HTML และ CSS ตามที่ออกแบบไว้ ดังรูปที่ 3.20



รูปที่ 3.20 ไฟล์ HTML และ CSS

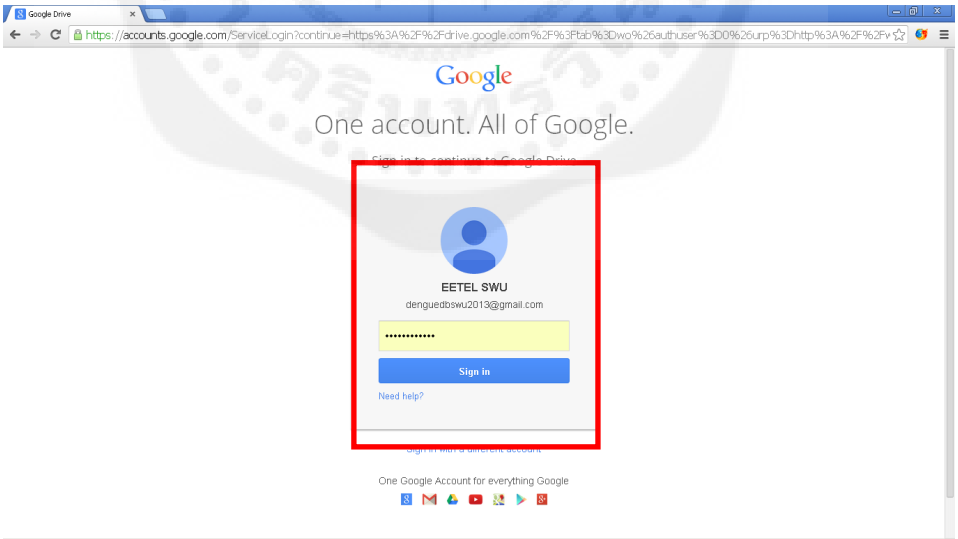
*หมายเหตุ ในกรณีที่ต้องใช้รูปภาพแสดงบนหน้าเว็บ ตั้งค่าระบุโฟลเดอร์ที่เก็บไฟล์รูปภาพที่ STATICFILES_DIRS ใน settings.py ดังรูปที่ 3.21



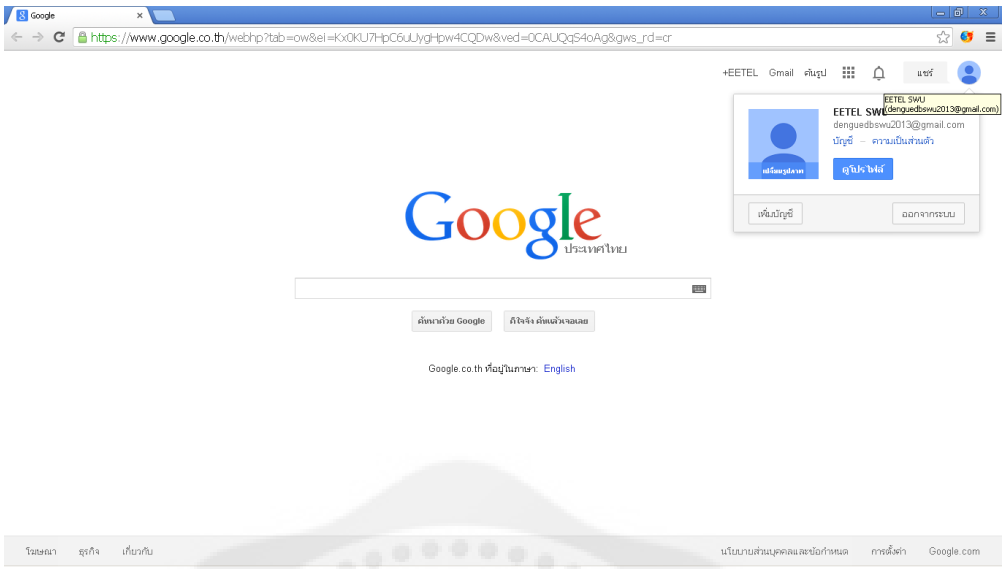
รูปที่ 3.21 ตั้งค่า STATICFILES_DIRS

3. วิธีสร้างเว็บแอพพลิเคชันด้วย Google apps script

3.1 ล็อกอินเข้า Google โดย Google account พิมพ์ชื่อและรหัสลงไปแล้วกด Sign in ดังรูปที่ 3.22

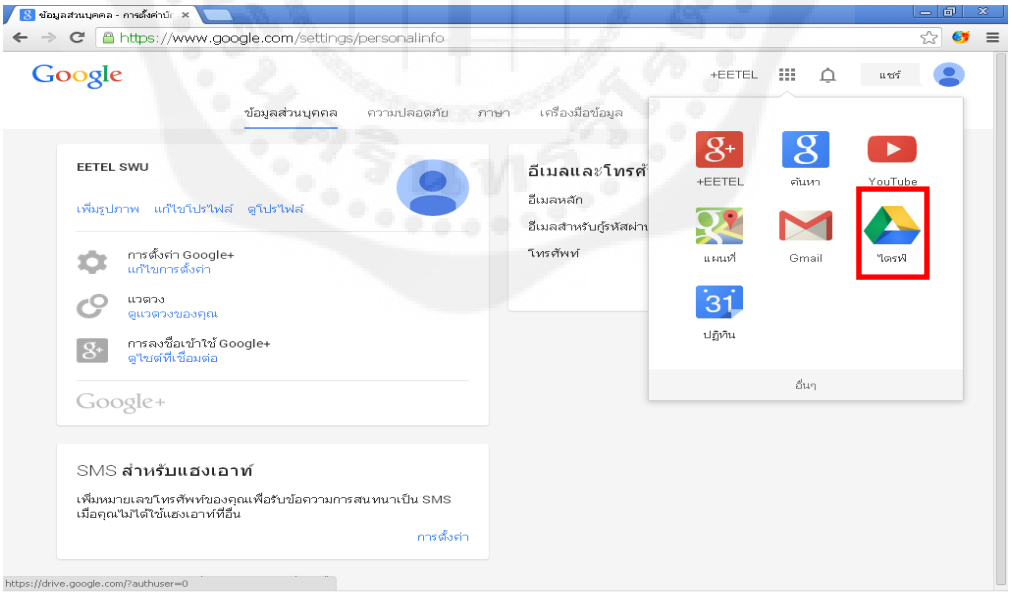


รูปที่ 3.22 ล็อคอินเข้าใช้งาน Google apps



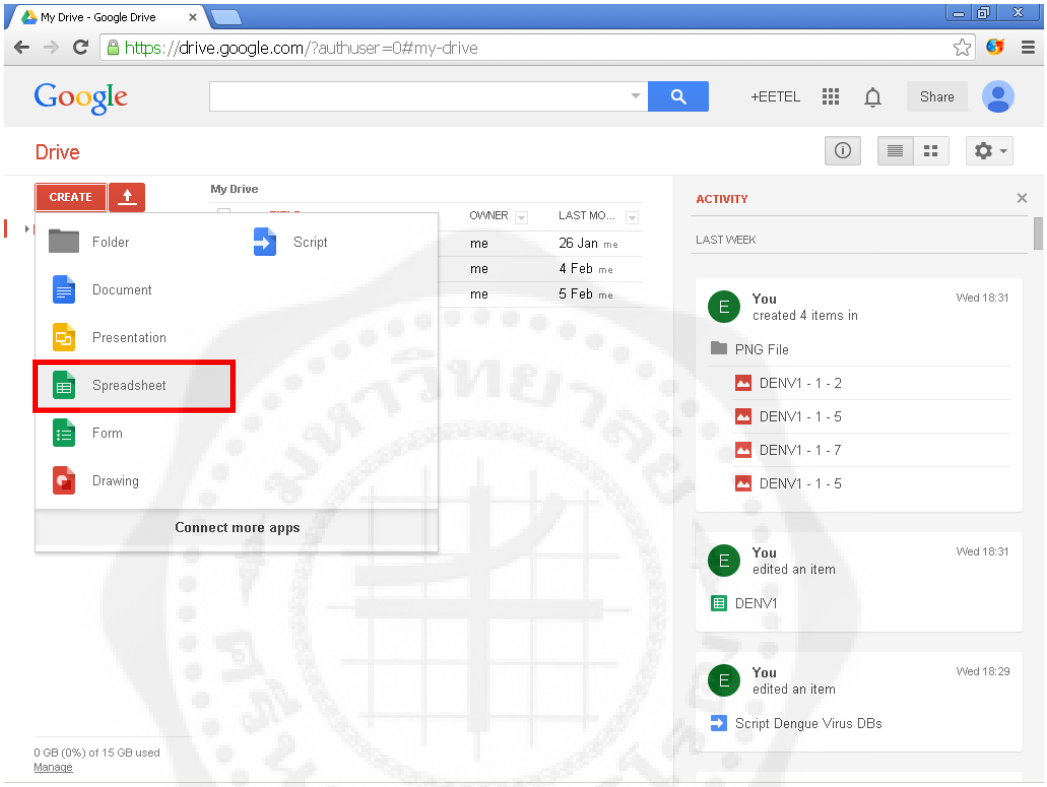
รูปที่ 3.23 ล็อคอิน Google account สำเร็จ

3.2 ใช้งาน Google drive ในการเก็บข้อมูล โดยการ Import ข้อมูลลงใน Spreadsheet ให้เข้าไปที่ Google drive ดังรูปที่ 3.24



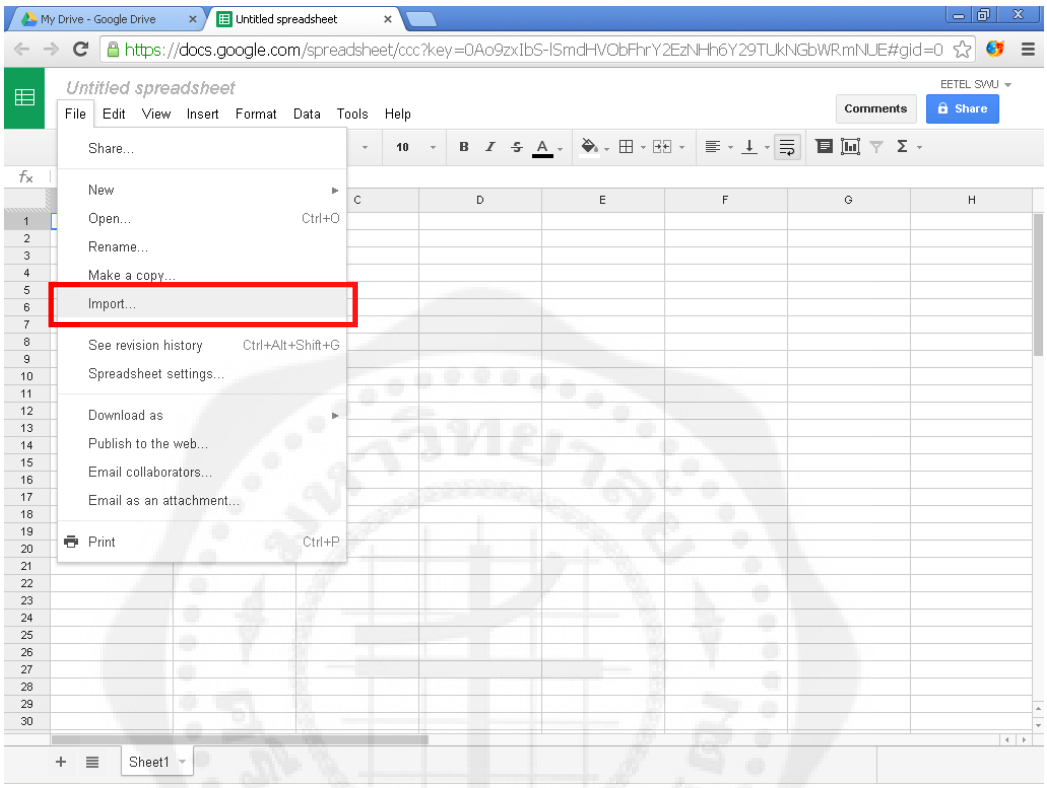
รูปที่ 3.24 การเข้าใช้งาน Google drive

3.3 ใช้ Google drive ในการจัดการไฟล์ฐานข้อมูล โดย Import ข้อมูลไฟล์ตาราง Tab-delimited ที่ต้องการ อัพโหลดขึ้นไว้ใน Google spreadsheet โดยการ Import ข้อมูลลงใน Spreadsheet ให้ทำการสร้าง Spreadsheet ดังรูปที่ 3.25



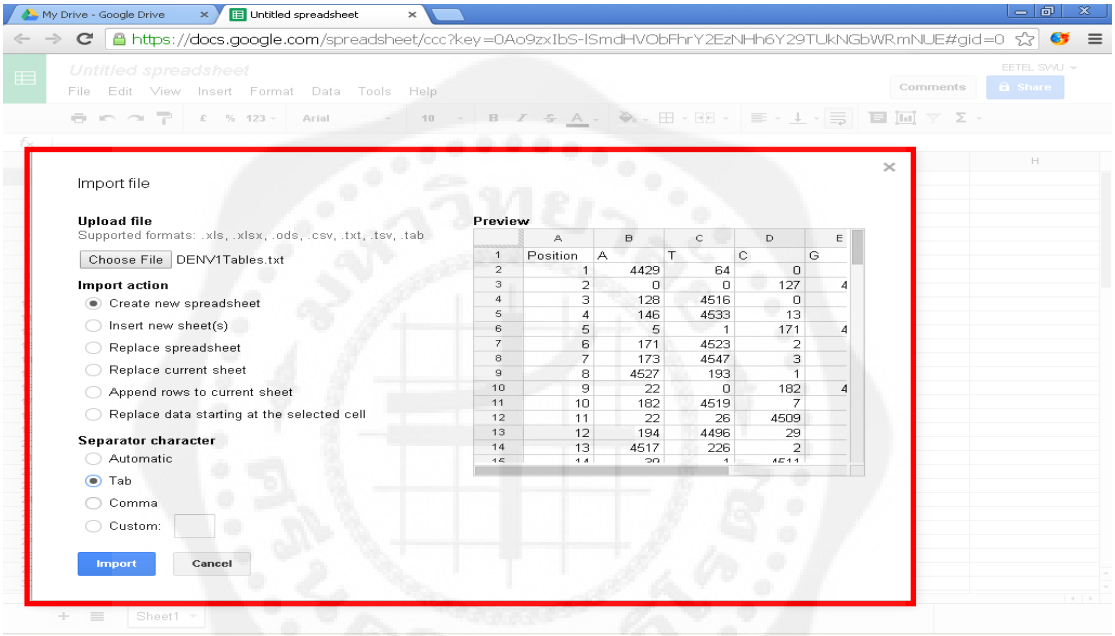
รูปที่ 3.25 สร้าง Spreadsheet ใน Google drive

3.4 จะเป็นเว็บเพจที่เหมือนกับหน้าตาโปรแกรม Microsoft excel ให้เราคลิกที่ File และคลิกที่ Import… ดังรูปที่ 3.26



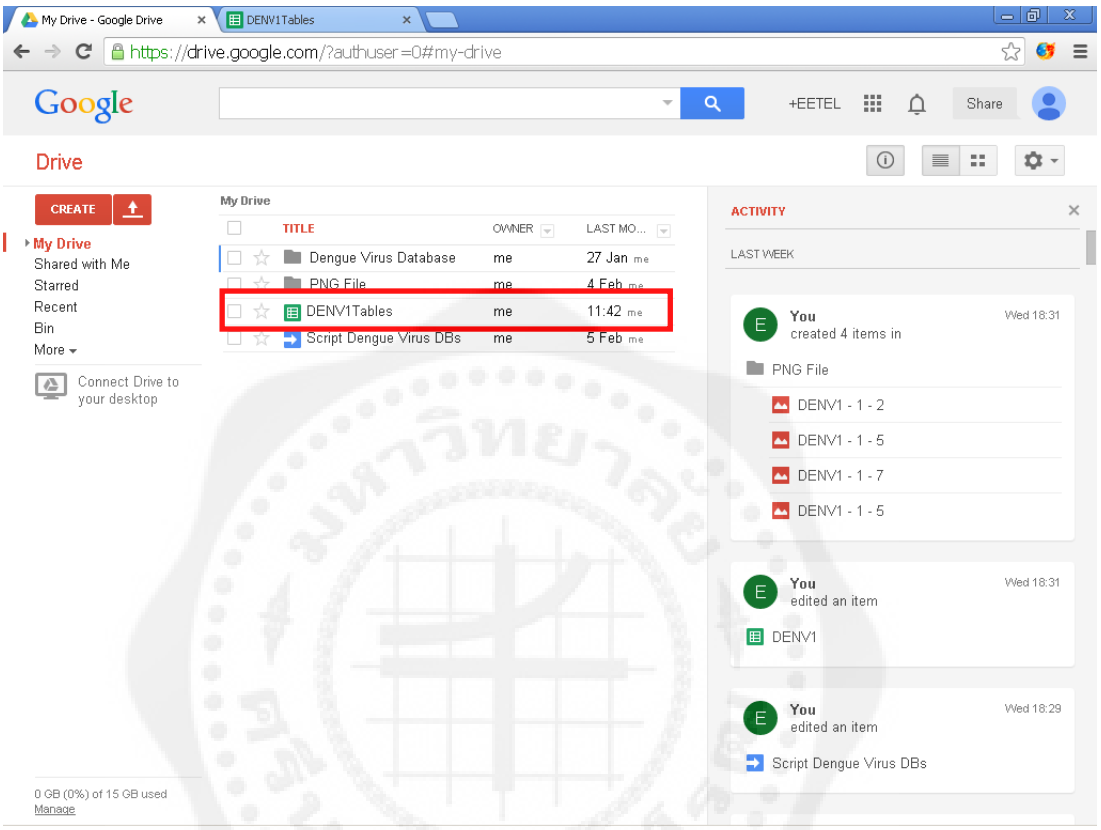
รูปที่ 3.26 Import File ลงบน Google spreadsheet (1)

3.5 คลิก Choose File เลือกไฟล์ที่เราต้องการอัพโหลดไฟล์ข้อมูลลงในGoogle spreadsheet ขึ้นมา ซึ่งจะรับไฟล์ประเภท .xls .xlsx .ods .csv .txt .tsv .tab ตัวอย่างเช่น ถ้าเราต้องการ Import ไฟล์ Tab-Delimited Import ชื่อ DENV1Tables.txt ได้โดยในส่วน Upload File คลิกที่ Choose File DENV1Tables.txt แล้ว Open ขึ้นมาจะได้ภาพตัวอย่างไฟล์ (Preview) ขึ้นอยู่ทางขวาของหน้าต่างส่วนของ Import Action ให้เลือกแบบ Create new spreadsheet และส่วนของ Separator Character เลือกแบบ Tab แล้วคลิก Import ดังรูปที่ 3.27

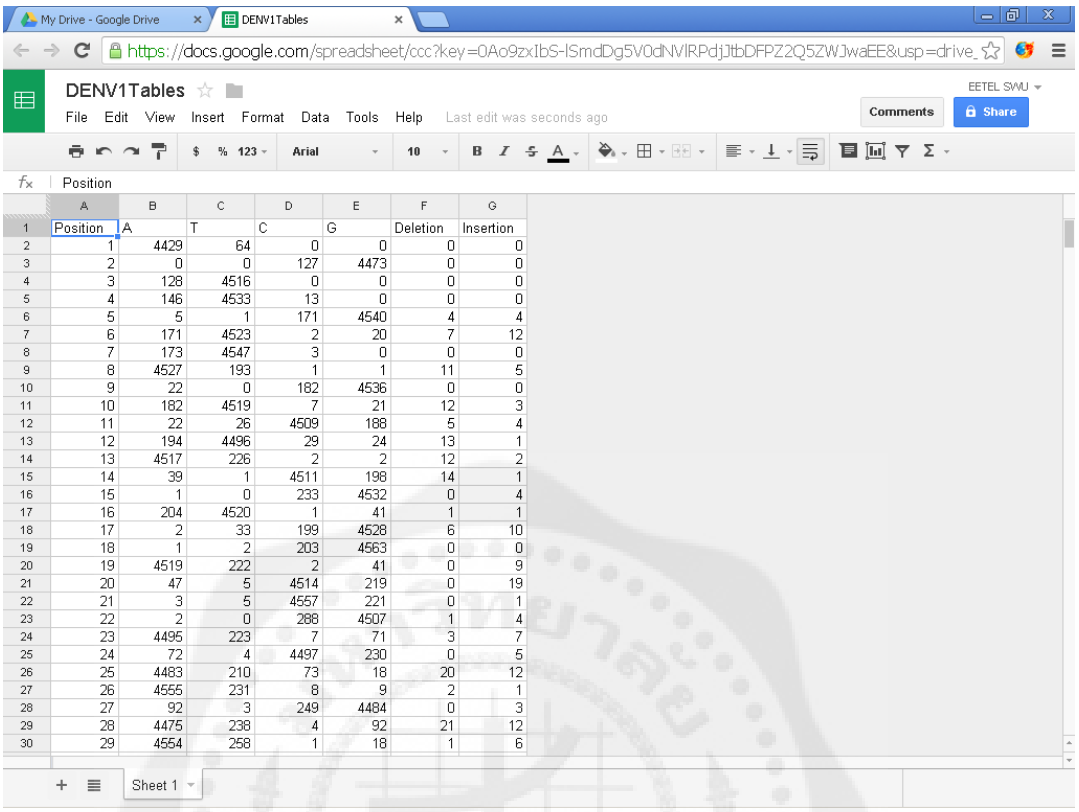


รูปที่ 3.27 Import file ลงบน Google spreadsheet (2)

3.6 จะได้ข้อมูลที่ต้องการไว้ใน Google spreadsheet อยู่บน Google drive ดังรูปที่ 3.28 และรูปที่ 3.29



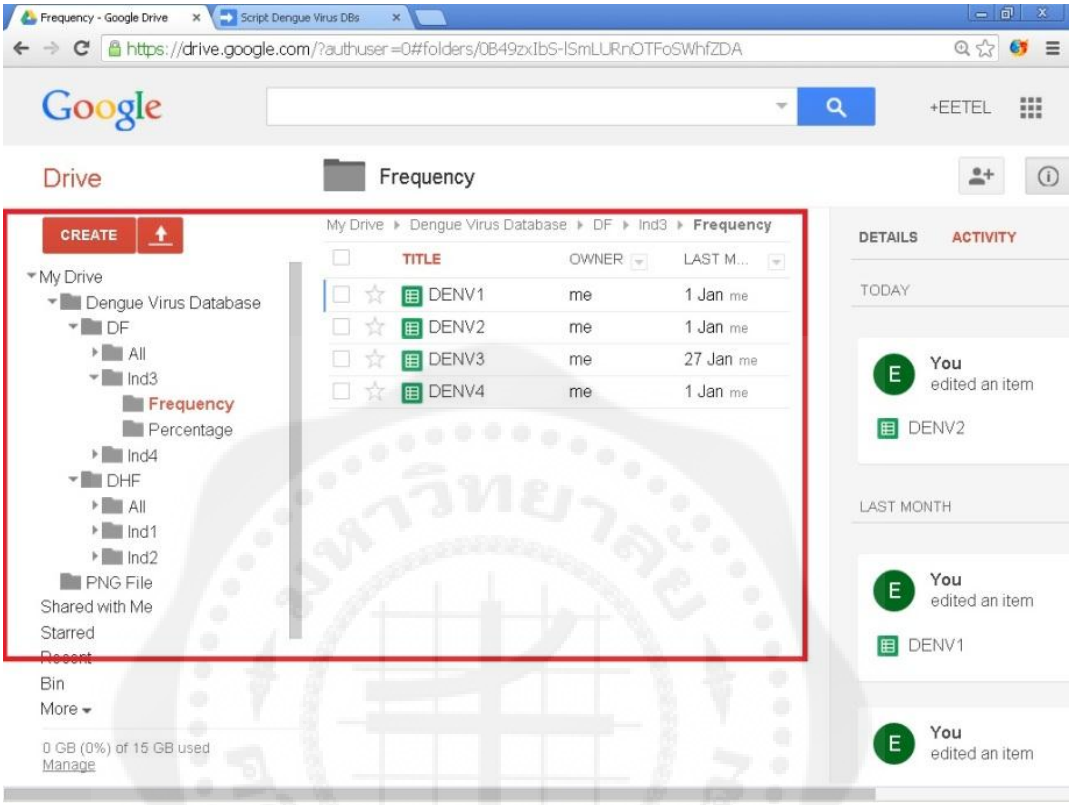
รูปที่ 3.28 ไฟล์ที่ Import ลงบน Google drive



	A	B	C	D	E	F	G
1	Position	A	T	C	G	Deletion	Insertion
2	1	4429	64	0	0	0	0
3	2	0	0	127	4473	0	0
4	3	128	4516	0	0	0	0
5	4	146	4533	13	0	0	0
6	5	5	1	171	4540	4	4
7	6	171	4523	2	20	7	12
8	7	173	4547	3	0	0	0
9	8	4527	193	1	1	11	5
10	9	22	0	182	4536	0	0
11	10	182	4519	7	21	12	3
12	11	22	26	4509	188	5	4
13	12	194	4496	29	24	13	1
14	13	4517	226	2	2	12	2
15	14	39	1	4511	198	14	1
16	15	1	0	233	4532	0	4
17	16	204	4520	1	41	1	1
18	17	2	33	199	4528	6	10
19	18	1	2	203	4563	0	0
20	19	4519	222	2	41	0	9
21	20	47	5	4514	219	0	19
22	21	3	5	4557	221	0	1
23	22	2	0	288	4507	1	4
24	23	4495	223	7	71	3	7
25	24	72	4	4497	230	0	5
26	25	4483	210	73	18	20	12
27	26	4555	231	8	9	2	1
28	27	92	3	249	4484	0	3
29	28	4475	238	4	92	21	12
30	29	4554	258	1	18	1	6

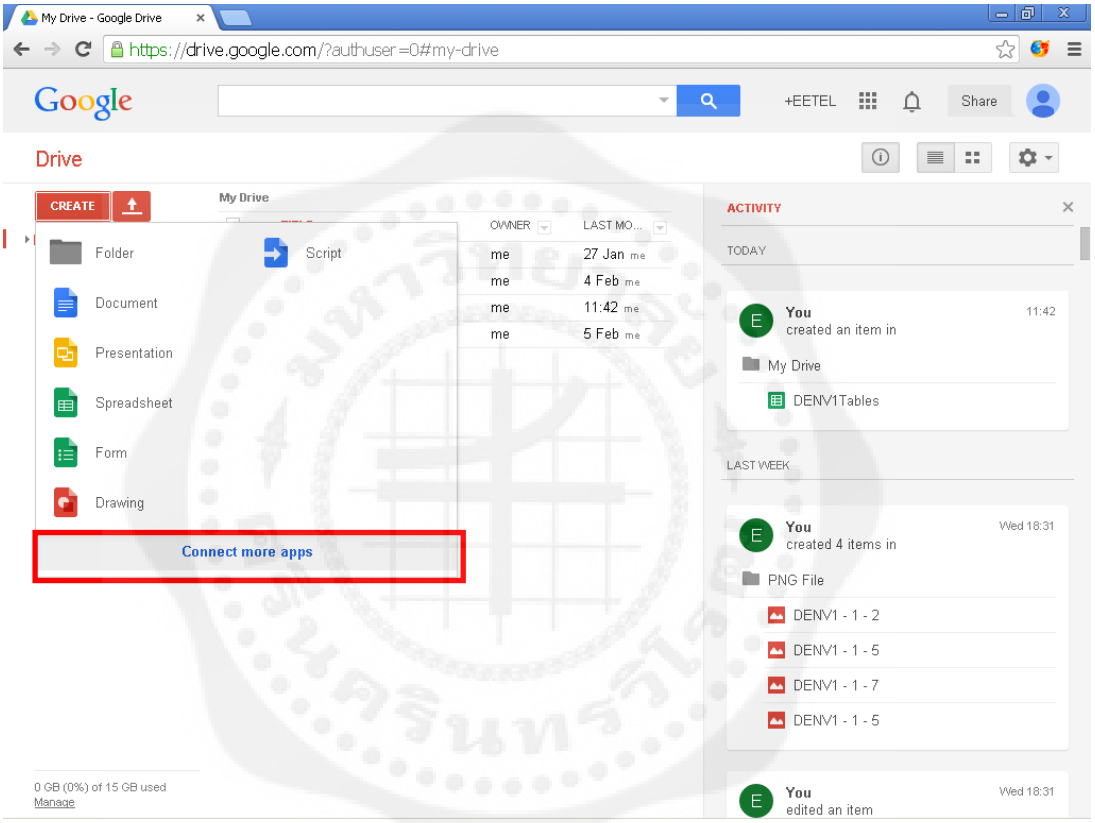
รูปที่ 3.29 ข้อมูลใน Google spreadsheet

3.7 เมื่อจัดการอัพโหลดเสร็จสิ้นแล้วก็จะได้ ดังรูปที่ 3.30



รูปที่ 3.30 การจัดการไฟล์ฐานข้อมูลบน Google drive

3.8 สร้าง Google apps script ซึ่งจะใช้ในการสร้าง Web application สามารถสร้าง Script ขึ้นมาโดยสังเกตในส่วนของ Google drive ที่ปุ่มสร้าง (Create) จะ มีบริการ Google apps ต่างๆ ซึ่งตัว Google apps script นั้นเราสามารถเรียกใช้งานได้จากตรงนี้ ในกรณีใช้งานครั้งแรกบน Google drive อาจจะไม่ปรากฏ Google apps script ให้ใช้งาน ต้องเรียกใช้งานโดย ให้คลิกที่ปุ่ม Create และคลิกที่ Connect more apps ดังรูปที่ 3.31

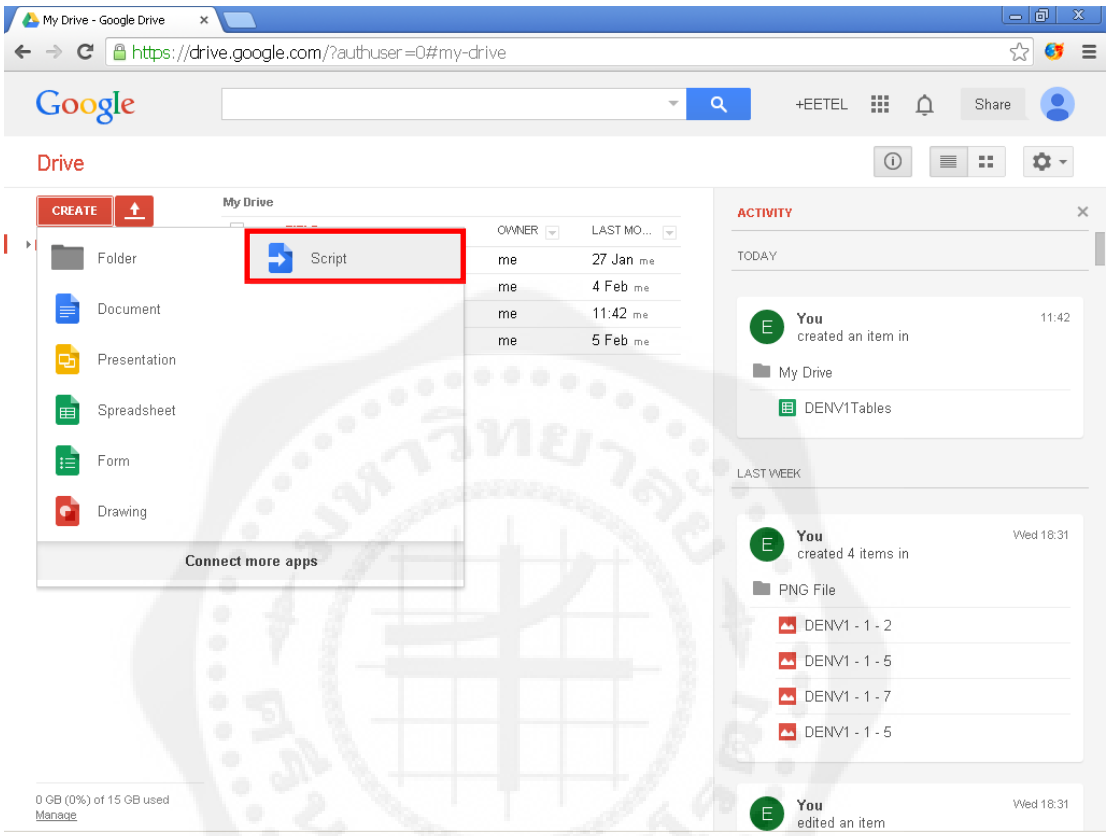


รูปที่ 3.31 การเรียกใช้ Google apps script เมื่อ Google drive ไม่ปรากฏการใช้งาน

3.8.1 ในส่วนนี้มีจะมีบริการของ Google apps มากมาย ให้เราหา Google apps script ตรงช่อง Search apps แล้วให้ คลิกปุ่ม Connect พอเชื่อมต่อสำเร็จตรงปุ่ม Connect นั้นจะเปลี่ยนเป็นคำว่า Rate it เราก็จะได้ Google apps script มาใช้งานบน Google drive ของเราเอง ดังรูปที่ 3.32

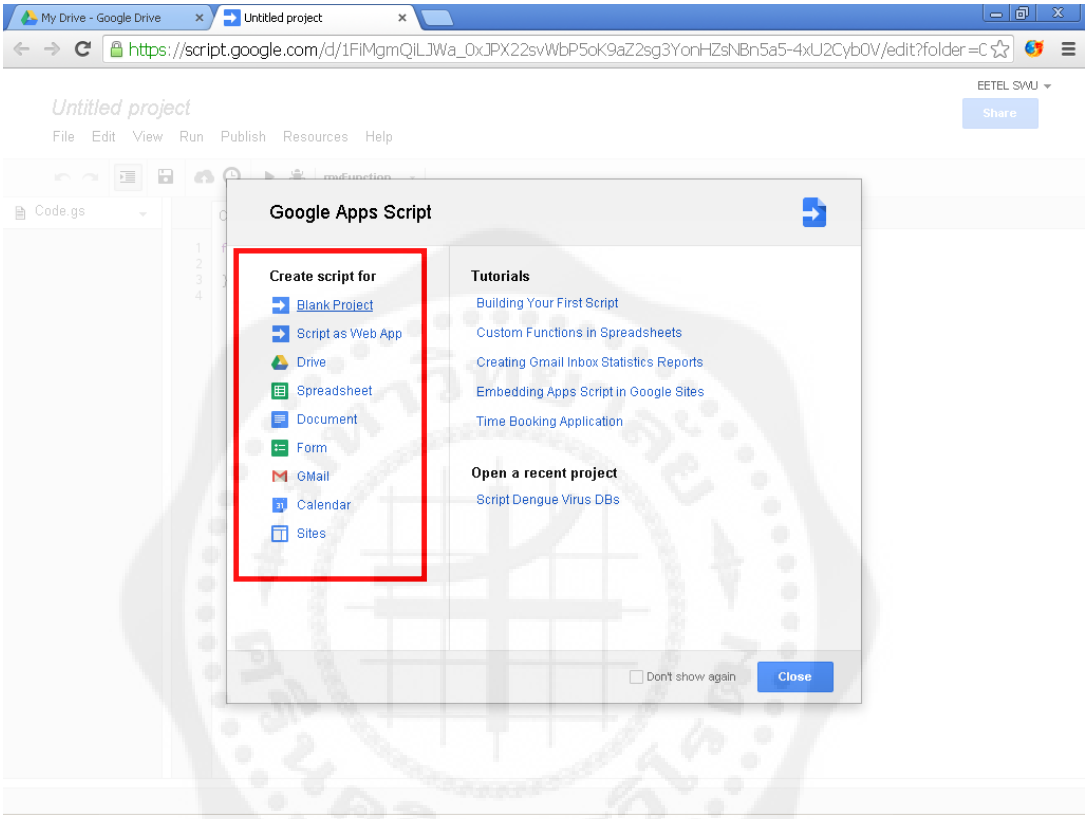
รูปที่ 3.32 เชื่อมต่อ Google apps script บน Google drive

3.9 ในการสร้าง Google apps script ทำได้โดยคลิกปุ่ม Create และเลือก Script ดังรูปที่ 3.33



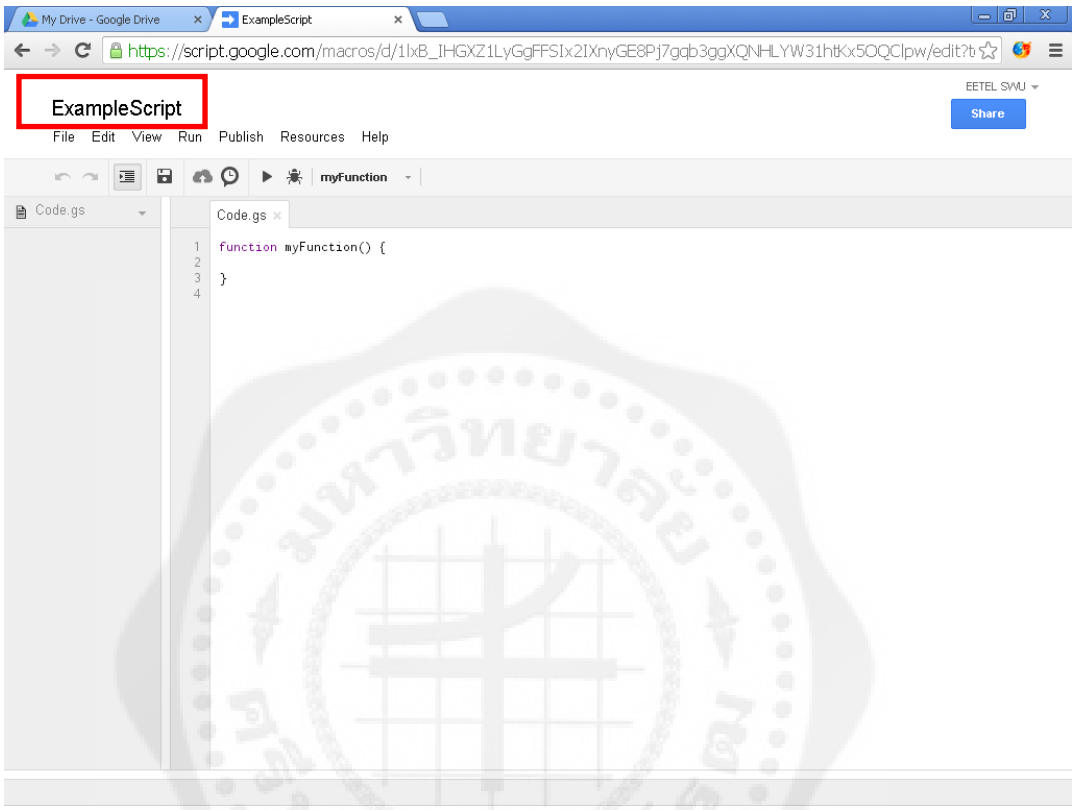
รูปที่ 3.33 การใช้งาน Google apps script บน Google drive

3.10 จะปรากฏหน้าต่าง Google apps script ขึ้นมา ตรงส่วน Create script for จะเป็นตัวเลือกในการสร้าง Script แบบต่างๆ ตัวอย่างเช่นเราจะสร้าง script เปล่าๆ ขึ้นมา 1 ไฟล์ ให้เลือกในส่วน Create script for เป็น Blank Project ดังรูปที่ 3.34

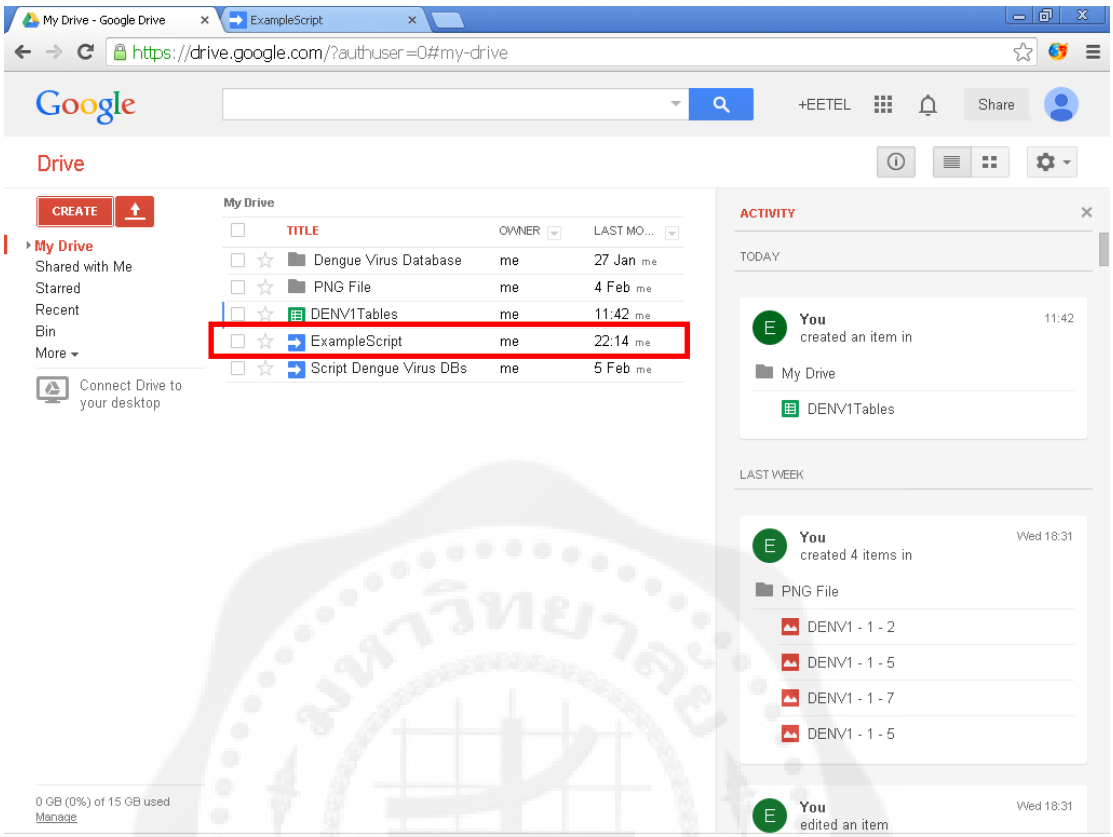


รูปที่ 3.34 การสร้าง Google apps script

3.11 จะขึ้นหน้าต่าง Script ทำการตั้งชื่อตรง Untitled Project ก็จะได้ไฟล์ Script มาใช้งาน เช่น ตั้งชื่อเป็น ExampleScript ดังรูปที่ 3.35 และรูปที่ 3.36

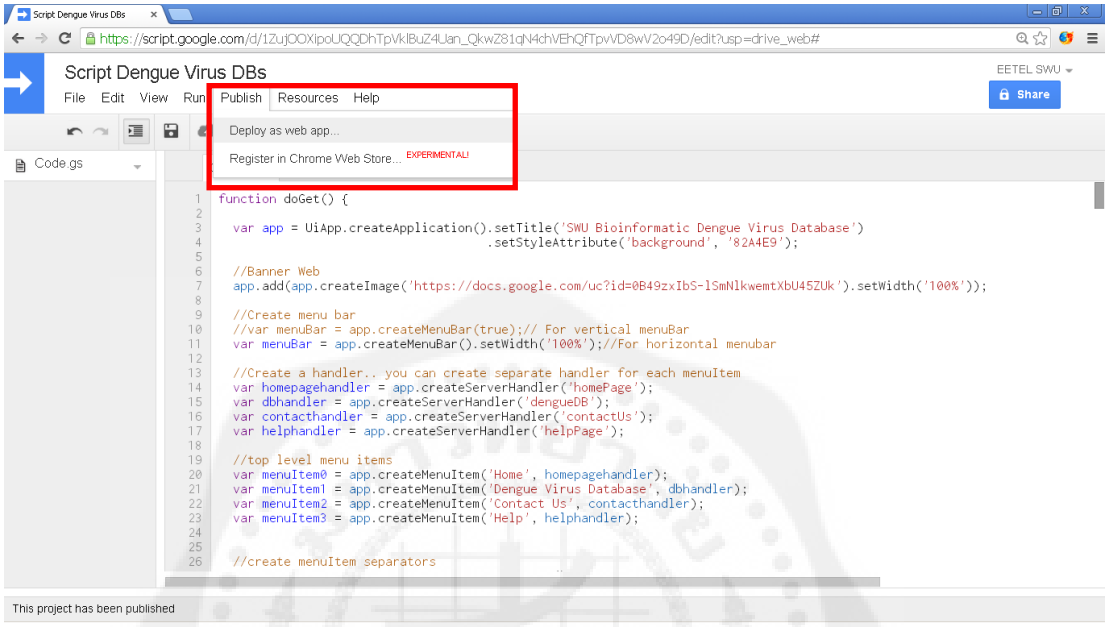


รูปที่ 3.35 โปรแกรม Google apps script



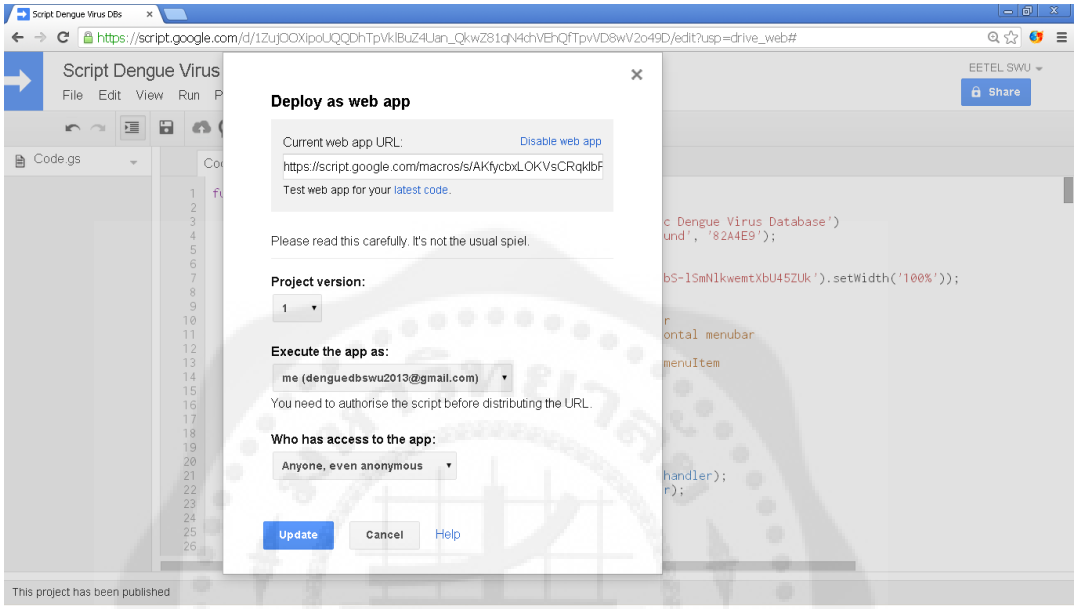
รูปที่ 3.36 ไฟล์ Script ที่สร้างขึ้น

3.12 ปรับ Script เป็นเว็บแอพพลิเคชัน (Deploy as web app) โดยเปิดไฟล์ Google app script ที่ต้องการขึ้นมา ให้คลิกที่แถบเมนูชื่อว่า Publish และ Deploy as web app… ดังรูปที่ 3.37



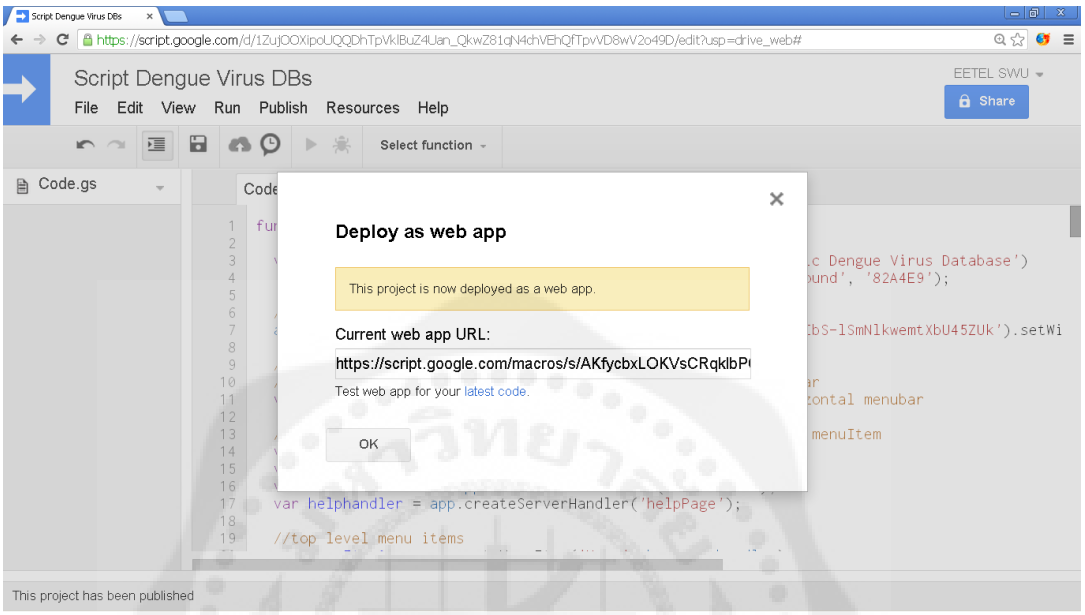
รูปที่ 3.37 Deploy as web app

3.13 จะปรากฏหน้าต่าง ดังรูปที่ 3.38 ให้ตั้งค่าดังนี้ ตรงส่วน Project version: ให้คลิก Manage version และตรงส่วนที่เป็น Who has access to the app: ให้ตั้งค่าเป็น Anyone, even anonymous เพื่อให้คอนนอกเข้ามาใช้งาน URL ของเราได้ แล้วคลิก Update



รูปที่ 3.38 การตั้งค่าเว็บแอพพลิเคชัน

3.14 จะปรากฏหน้าต่างดังรูปที่ 3.39 เราก็จะได้ URL ของเว็บแอพพลิเคชันเราและเว็บแอพพลิเคชันที่เราสร้างขึ้นด้วย Google apps script



รูปที่ 3.39 URL เว็บแอพพลิเคชัน

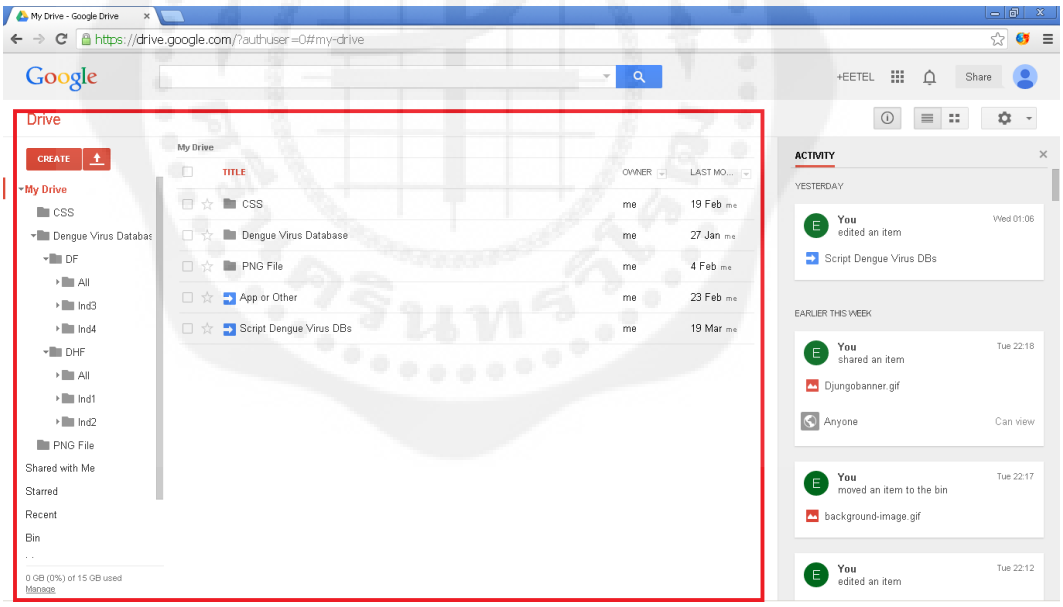
บทที่ 4
ผลการทดลอง

หลังจากได้ทดลองสร้างเว็บแอพพลิเคชันเรียบร้อยแล้ว เว็บแอพพลิเคชันสามารถทำงานได้ตามวัตถุประสงค์ของงาน ดังนี้

1. ผลการเปรียบเทียบ Google apps script กับ Django

1.1 การสำรองข้อมูล (Google apps script)

ไม่ต้องสำรองข้อมูลเองเนื่องจาก ใช้ Cloud ของ Google เป็นที่จัดเก็บข้อมูล Google จะสำรองข้อมูลให้เราเองอัตโนมัติ



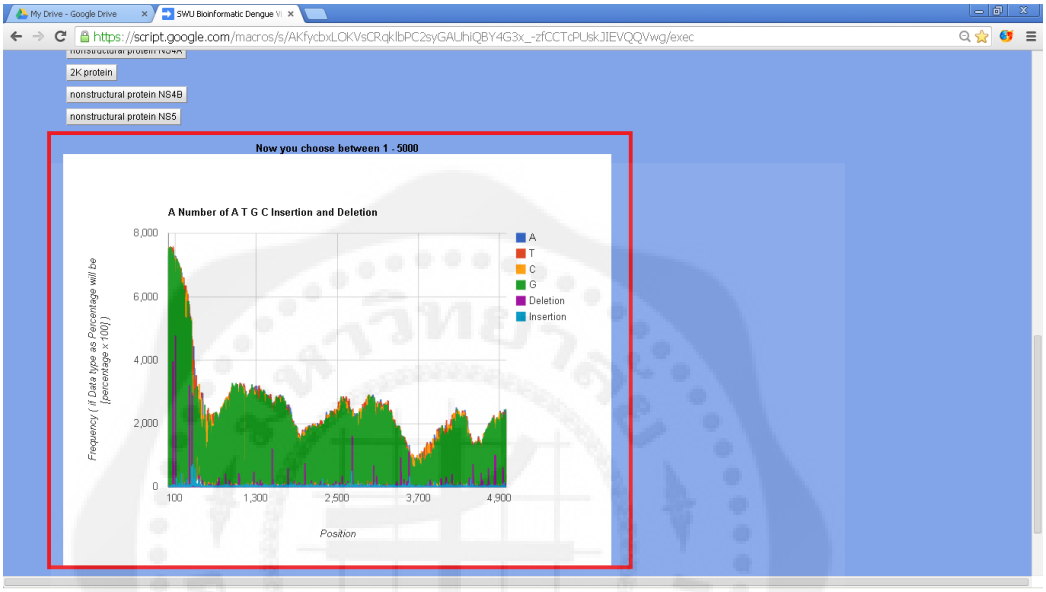
รูปที่ 4.1 การสำรองข้อมูลบน Google apps script

1.2 การสำรองข้อมูล (Django)

เนื่องจาก Django ต้อง Deploy ขึ้นไปบน Server ไฟล์อาจเสียได้ได้จึงต้องทำการสำรองข้อมูลและจัดการไฟล์ด้วยตัวเอง

1.3 ความเร็วในการประมวลผลกราฟ (Google apps script)

เนื่องจากการใช้งานผ่านอินเตอร์เน็ตและการฝากข้อมูลไว้บน Cloud ของ Google ทำให้การเรียกดูข้อมูลกราฟของค่าตำแหน่งที่เป็นจำนวนมาก ต้องใช้เวลานานมาก เช่น ต้องการเรียกดูตำแหน่งจาก 1 – 5000 ต้อรอเวลาประมาณ 5 วินาที กราฟถึงจะปรากออกมา



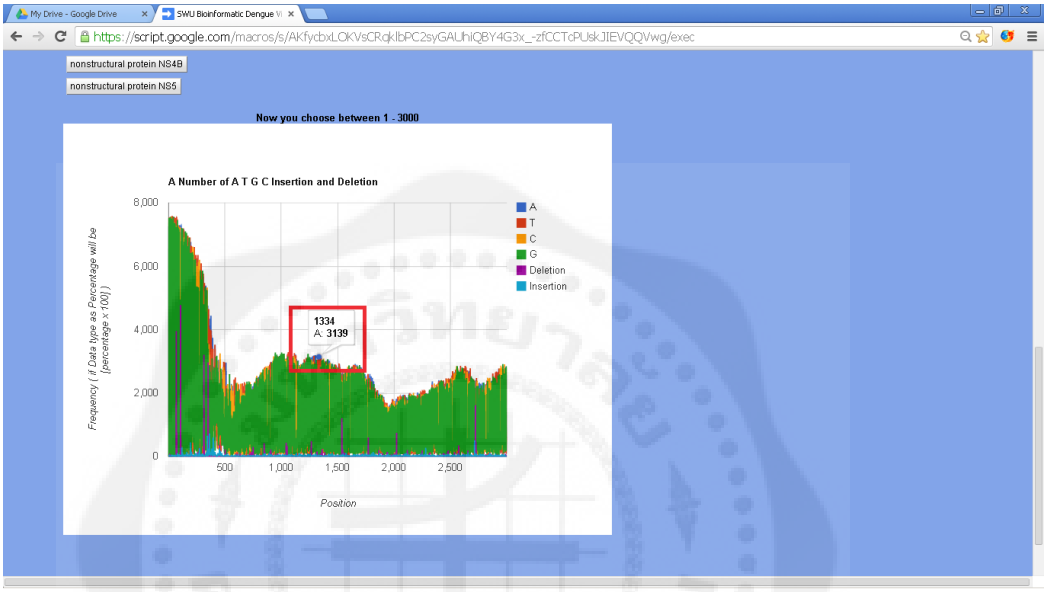
รูปที่ 4.2 การประมวลผลกราฟของ Google apps script

1.4 ความเร็วในการประมวลผลกราฟ (Django)

Django จะทำการประมวลผลผ่านโปรแกรมบน Server โดยไม่ต้องผ่านระบบอินเทอร์เนตแล้วจึงแสดงขึ้นบนเว็บเป็นไฟล์รูปความเร็วในการแสดงผลลัพธ์จึงขึ้นอยู่กับความเร็วของเครื่อง Server

1.5 การแสดงกราฟ (Google apps script)

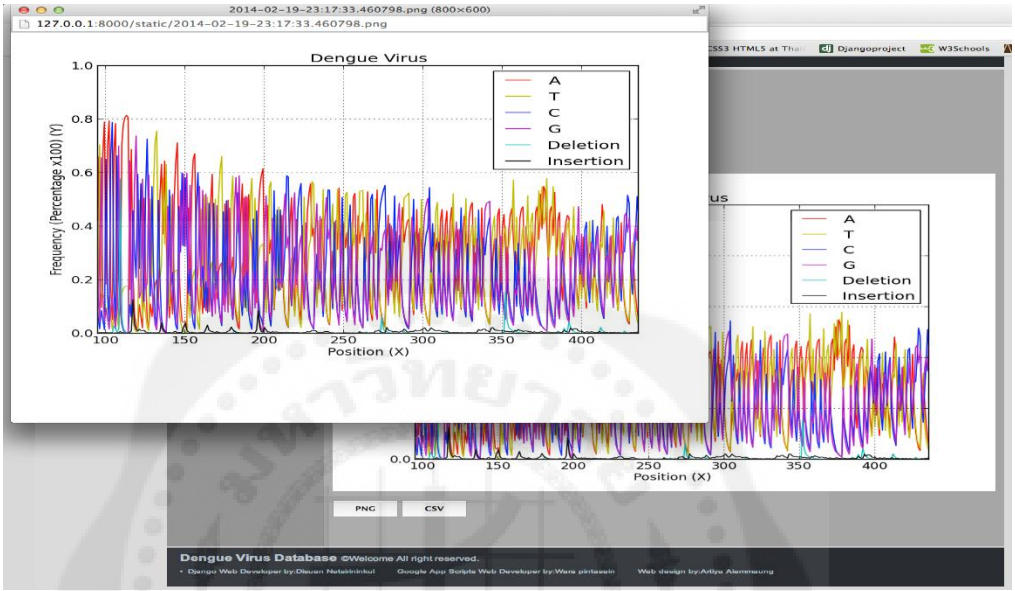
กราฟที่แสดงผลด้วย Google apps script สามารถชี้ดูข้อมูลผ่านกราฟที่แสดงขึ้นมานั้นได้ เช่น ชี้เส้นกราฟสีฟ้าตรงจุดแกน Position = 1334 ก็จะบอกค่าบนแกน Frequency ของ A = 3139



รูปที่ 4.3 การแสดงกราฟ Google apps script

1.6 การแสดงกราฟ (Django)

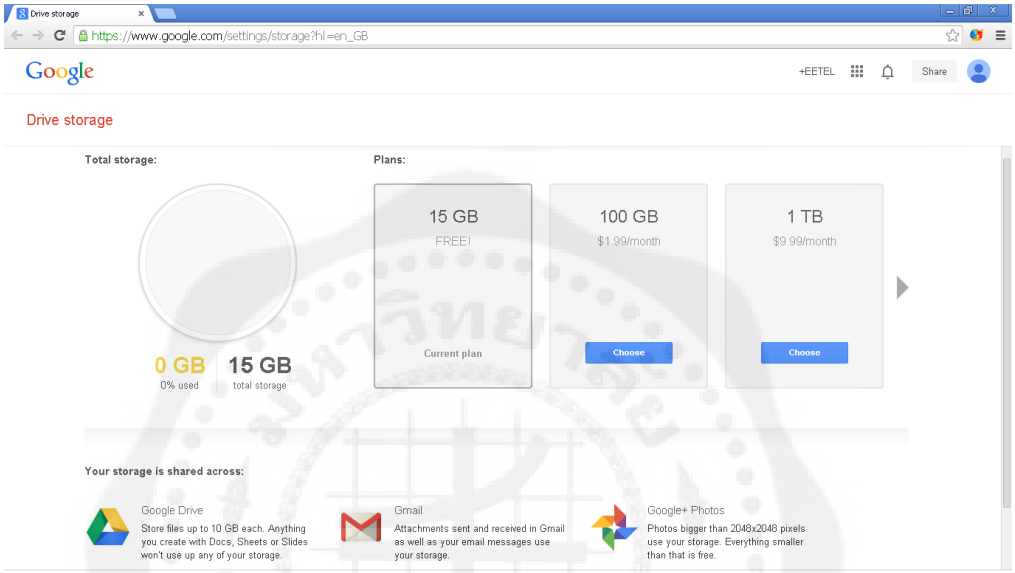
การแสดงผลกราฟของ Django จะแสดงออกมาในรูปแบบของไฟล์ png ซึ่งไม่สามารถที่จะชี้ดูตำแหน่งบนกราฟได้



รูปที่ 4.4 การแสดงกราฟ Django

1.7 ความสามารถในการจุข้อมูล (Google apps script)

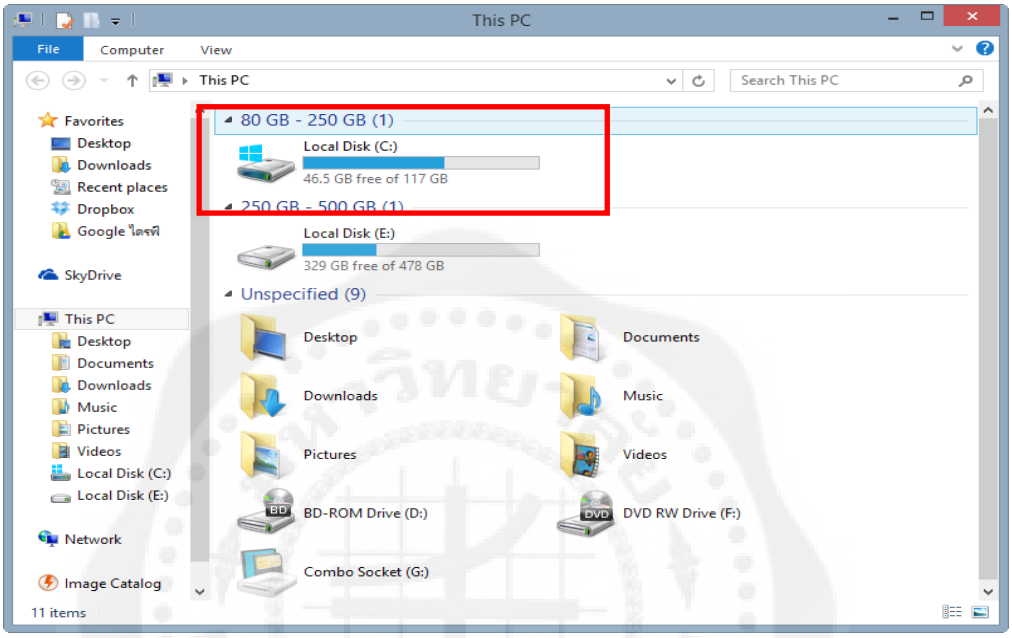
พื้นที่ในการจัดเก็บข้อมูลบน Cloud ของ Google สามารถใช้บริการได้ฟรีมีพื้นที่เพียงแค่ 15 GB ถ้าเราต้องการขยายพื้นที่จัดเก็บข้อมูลเพิ่มต้องเสียค่าใช้จ่ายในการใช้บริการให้กับบริษัท Google



รูปที่ 4.5 ความจุข้อมูลของ Google apps script

1.8 ความสามารถในการจุข้อมูล (Django)

ส่วนพื้นที่ที่ใช้ในการรองรับข้อมูลของ Django จะขึ้นอยู่กับพื้นที่ความจุของเครื่อง Server



รูปที่ 4.6 ความจุข้อมูลของ Django

2. ความสามารถของเว็บแอพพลิเคชัน

เว็บแอพพลิเคชันนี้เป็นเว็บไซต์เกี่ยวกับฐานข้อมูลเชื้อไวรัสไข้เลือดออก และเป็นการเปรียบเทียบความสามารถระหว่างสองแพลตฟอร์ม คือ Django กับ Google apps script ซึ่งผู้ใช้งานสามารถใช้งานได้ดังนี้

2.1 ผู้ใช้สามารถเลือกชนิดของโรคไข้เลือดออกได้คือระหว่าง Dengue Fever และ Dengue Hemorrhagic Fever ตามต้องการ

2.2 ผู้ใช้สามารถเลือกกลุ่มผู้ติดเชื้อได้จากโรคไข้เลือดออกทั้งสองชนิด คือ

2.2.1 Dengue Fever ได้แก่ Ind3 Ind4 และ All-DF

2.2.2 Dengue Hemorrhagic Fever ได้แก่ Ind1 Ind2 และ All-DH

2.3 ผู้ใช้จะสามารถเลือกพล๊อตกราฟได้ระหว่าง Frequency และ Percentage เพื่อแสดงเอาท์พุตออกมาในรูปของกราฟเชิงสถิติ

2.4 ผู้ใช้สามารถระบุตำแหน่งหรือช่วงของกราฟที่ผู้ใช้ต้องการได้

2.5 ผู้ใช้สามารถเลือกแสดงกราฟตามตำแหน่งของ Structural Proteins ได้

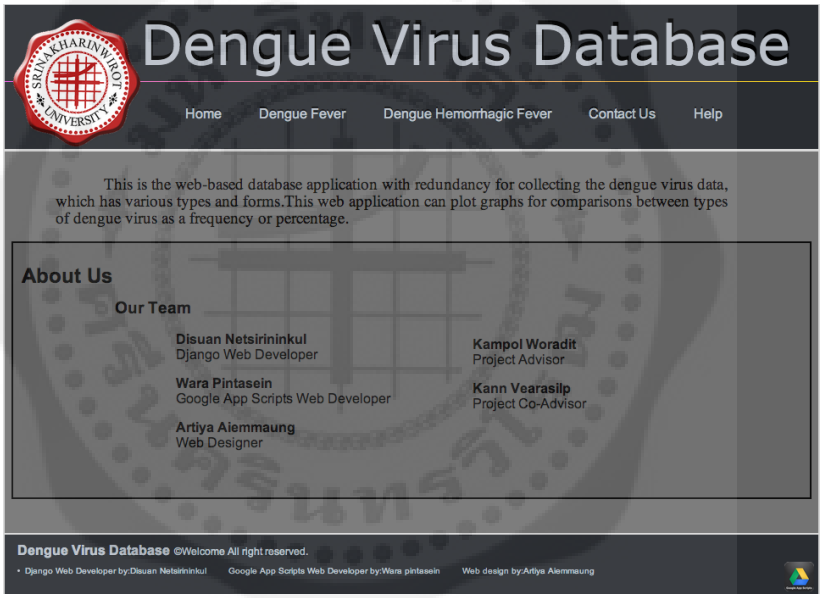
2.6 ผู้ใช้สามารถพล๊อตกราฟจากนั้นผู้ใช้สามารถบันทึกกราฟนั้นเป็นไฟล์ .png หรือบันทึกข้อมูลบนช่วงที่พล๊อตกราฟเป็นไฟล์ .csv ได้

2.7 ผู้ใช้สามารถคลิ๊กลิ้งค์เพื่อเข้าไปใช้งานในส่วนของ Google app scripts ได้

3. ส่วนของแพลตฟอร์มจังโก้

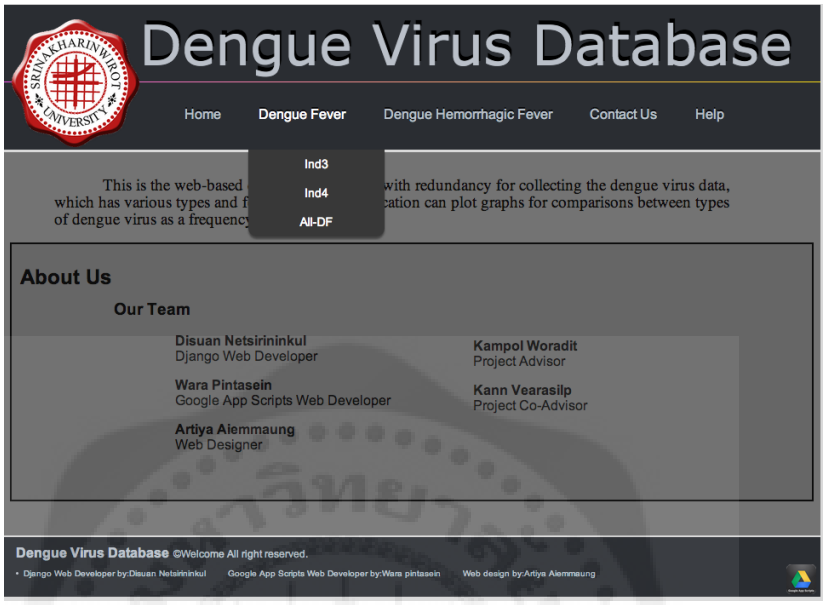
3.1 หน้า Home มีลักษณะดังรูปที่4.7ผู้ใช้สามารถเลือกกลุ่มผู้ติดเชื้อได้จากโรคไข้เลือดออกทั้งสองชนิด คือ

3.1.1 Dengue Fever ได้แก่ Ind3 Ind4 และ All-DF



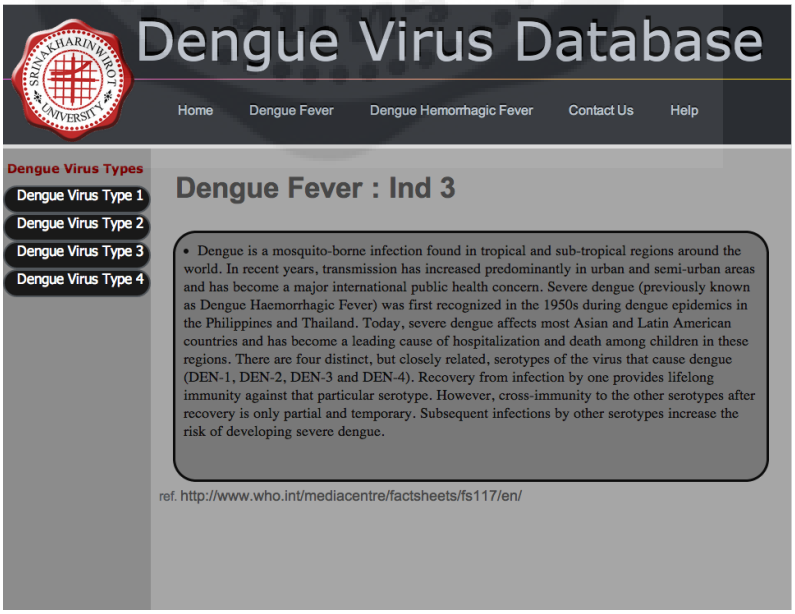
รูปที่ 4.7 หน้า Home

3.1.2 Dengue Hemorrhagic Fever ได้แก่ Ind1 Ind2 และ All-DH ดังรูปที่ 4.8

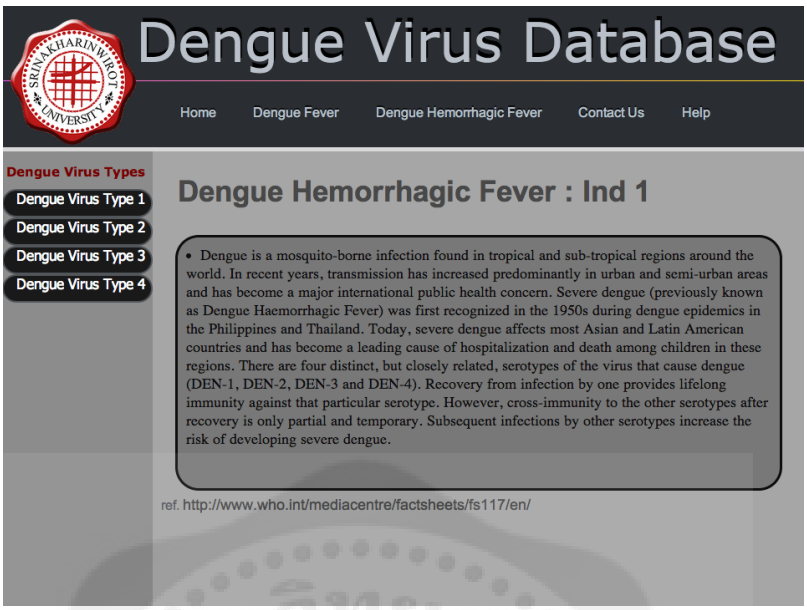


รูปที่ 4.8 Dengue Fever Menu

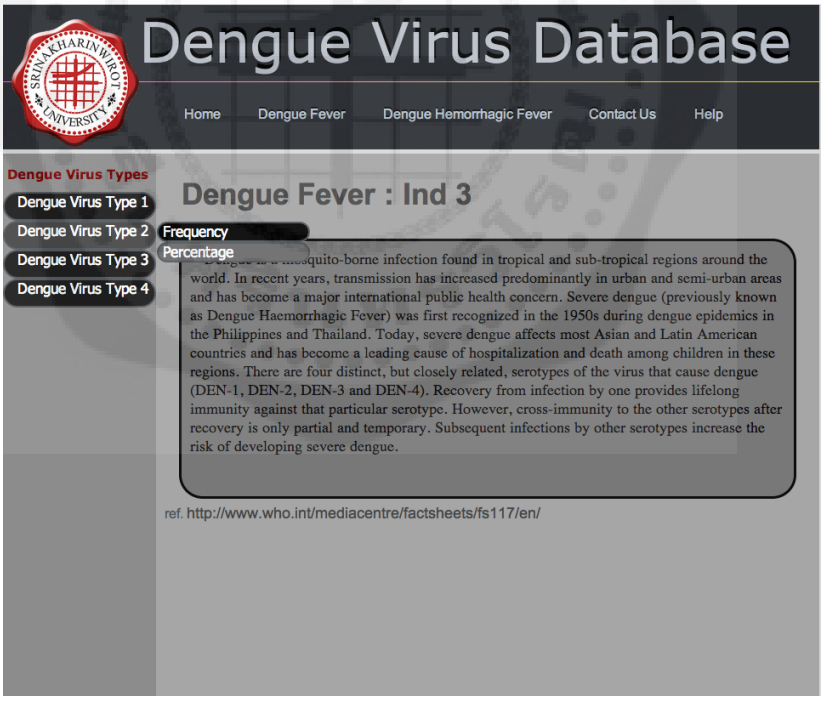
3.2 หน้า Dengue Fever และ Dengue Hemorrhagic Fever ดังรูปที่ 4.9 และ 4.10 หน้านี้ผู้ใช้งานสามารถเลือกพล๊อตกราฟได้ระหว่าง Frequency และ Percentage ดังรูปที่ 4.11



รูปที่ 4.9 หน้า Dengue Fever > Ind3

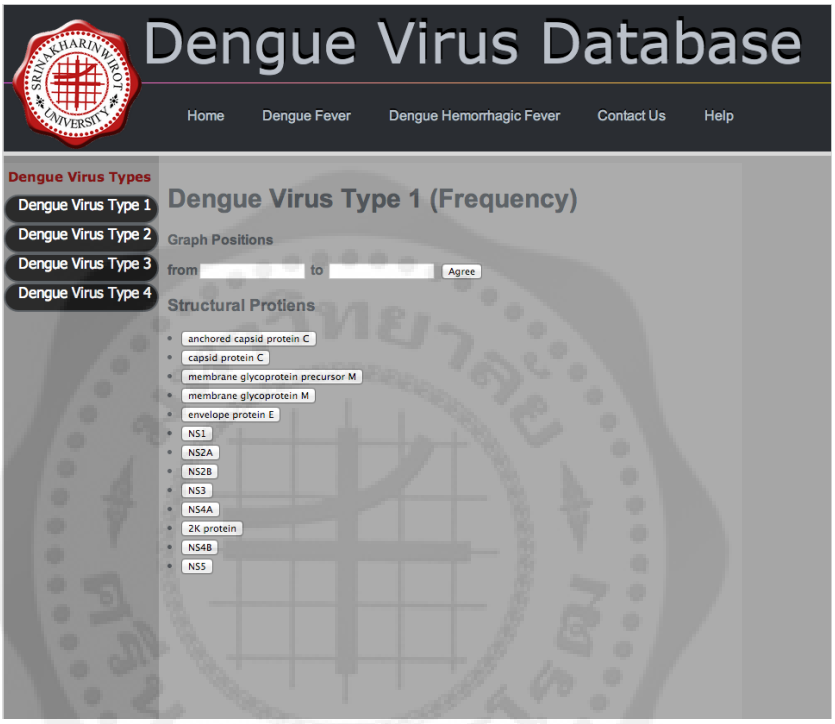


รูปที่ 4.10 หน้า Dengue Hemorrhagic Fever > Ind1

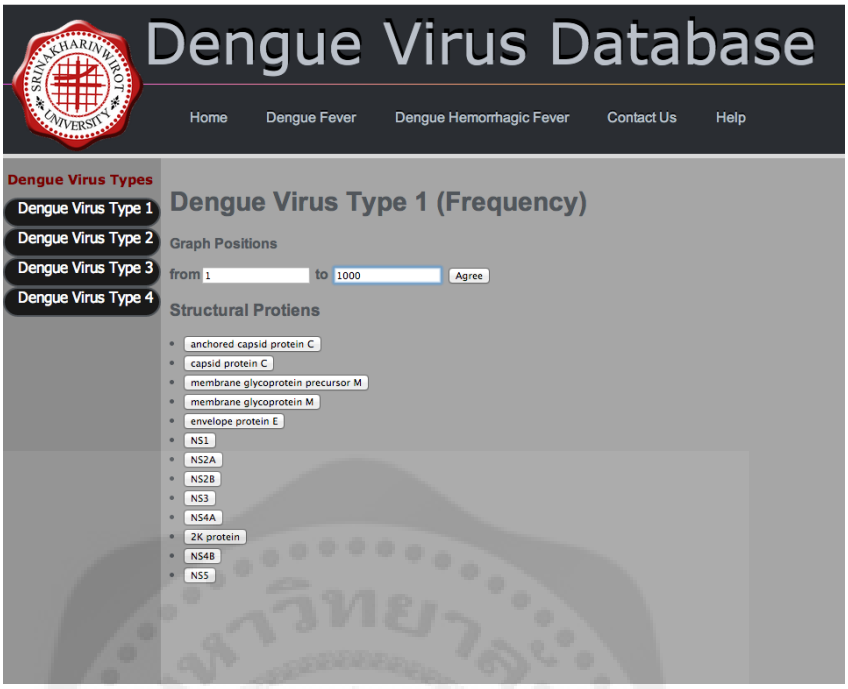


รูปที่ 4.11 การเลือกระหว่าง Frequency หรือ Percentage

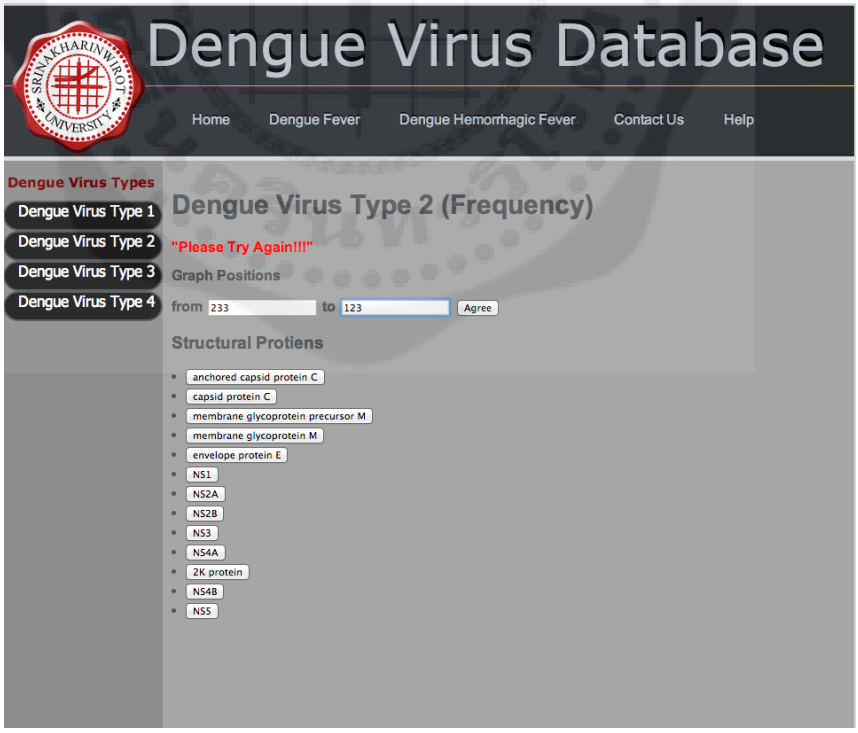
3.3 ดังรูปที่ 4.12 หน้านี้เป็นหน้าที่ผู้ใช้งานสามารถระบุตำแหน่งหรือช่วงของกราฟที่ผู้ใช้ต้องการได้ ดังรูปที่ 4.13 อีกทั้งยังสามารถเลือกแสดงกราฟตามตำแหน่งของ Structural Proteins ได้โดยคลิกที่ปุ่มชนิดของ Structural Proteins และถ้าผู้ใช้ใส่ค่าผิดรูปแบบจะมีคำเตือนให้ผู้ใช้ใส่ค่าใหม่ดังรูปที่ 4.14



รูปที่ 4.12 ช่องระบุตำแหน่งและปุ่ม Structural Proteins

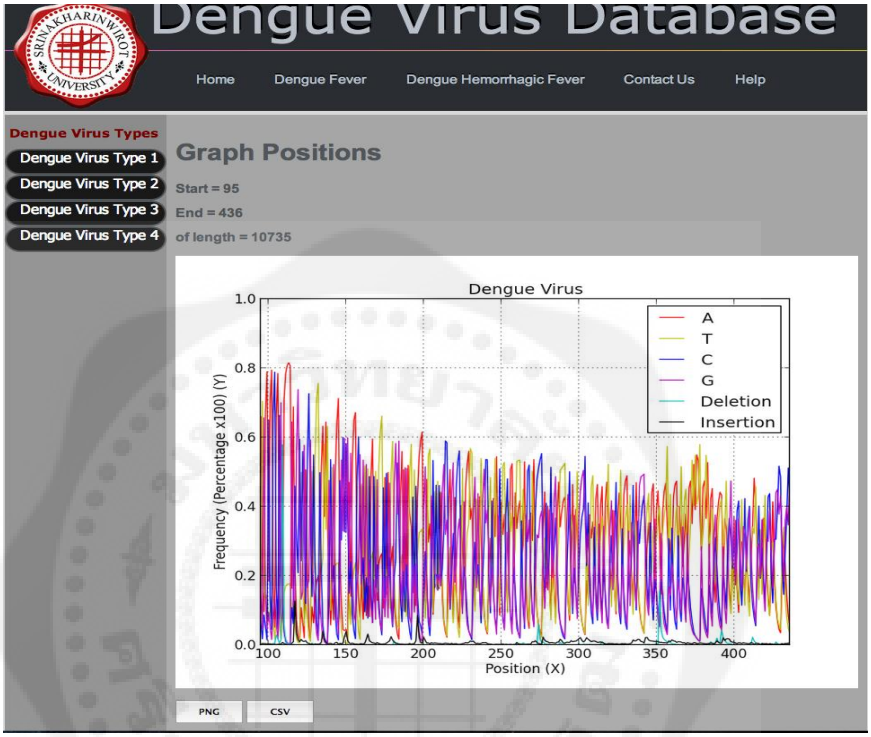


รูปที่ 4.13 การระบุตำแหน่งของกราฟ

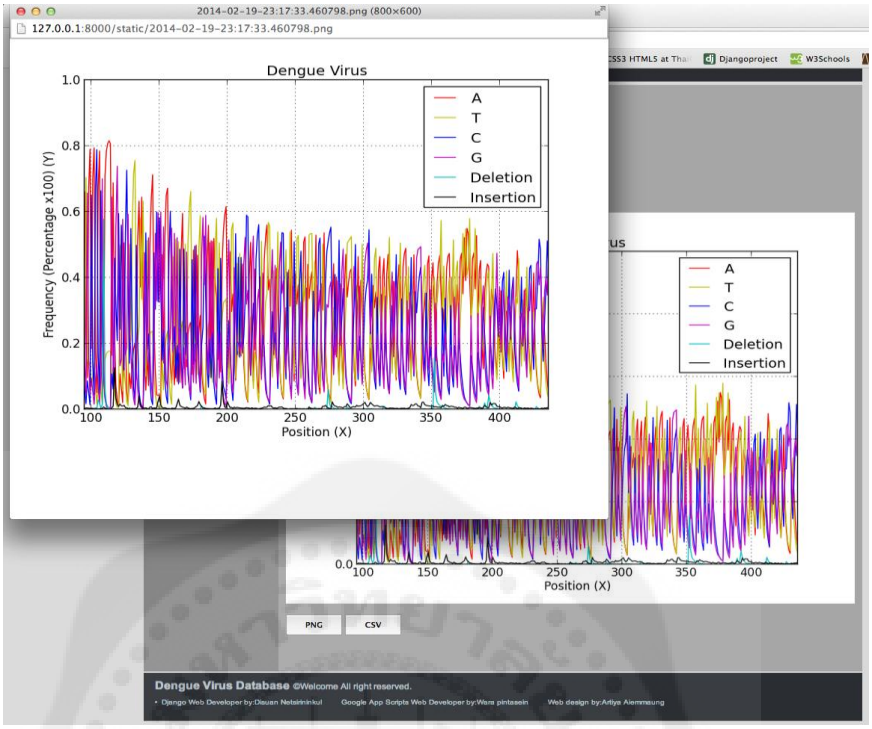


รูปที่ 4.14 แสดงข้อความเตือนเมื่อผู้ใช้ระบุตำแหน่งเริ่มต้นมากกว่าตำแหน่งสุดท้าย

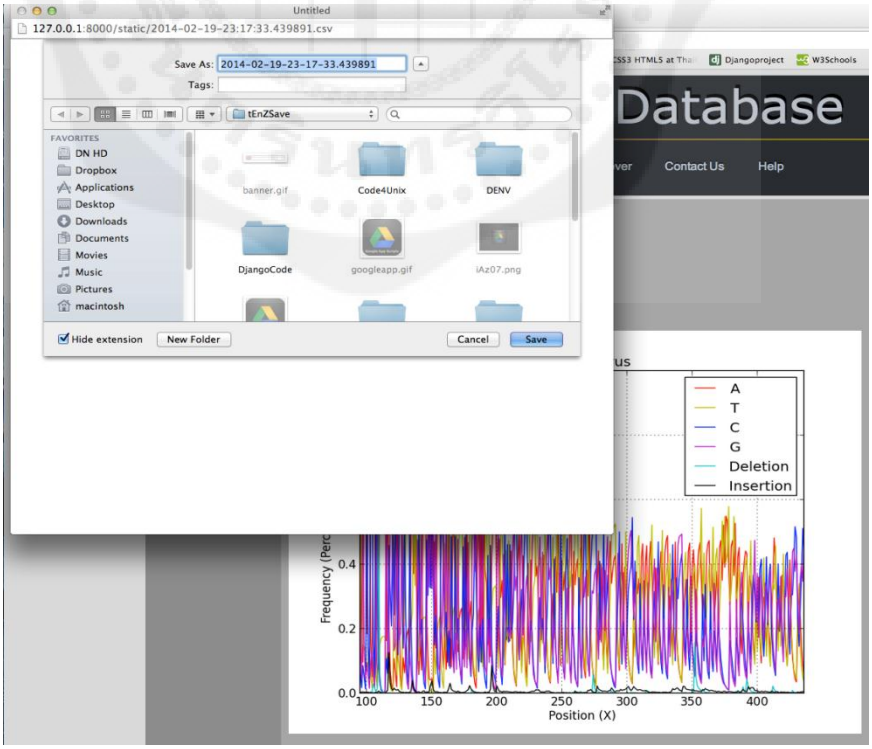
3.4 ผู้ใช้สามารถพล๊อตกราฟได้ดังรูปที่ 4.15 อีกทั้งผู้ใช้สามารถบันทึกกราฟนั้นเป็นไฟล์ .png หรือบันทึกข้อมูลบนช่วงที่พล๊อตกราฟเป็นไฟล์ .csv ได้เมื่อกดปุ่ม PNG และ CSV ดังรูปที่ 4.16 และ 4.17 ตามลำดับ



รูปที่ 4.15 แสดงกราฟตำแหน่งที่ 95 ถึง 436

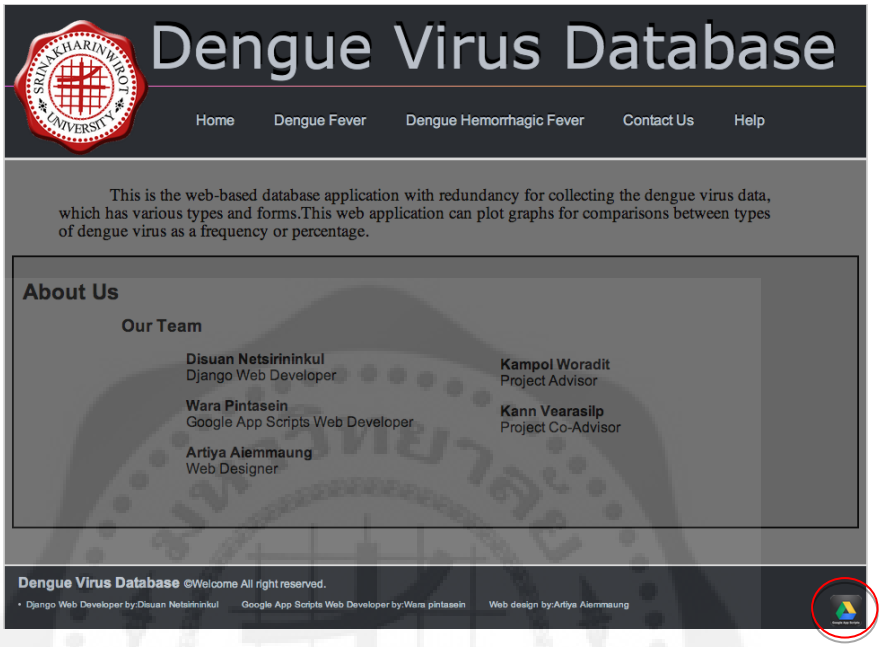


รูปที่ 4.16 หน้าต่างบันทึกกราฟเป็นไฟล์ .PNG



รูปที่ 4.17 หน้าต่างบันทึกข้อมูลบนกราฟเป็นไฟล์ .CSV

3.5 ผู้ใช้สามารถคลิ๊กลิ้งค์เพื่อเข้าไปใช้งานในส่วนของ Google apps script ได้จากปุ่มมุมขวาล่าง ดังรูปที่ 4.18



รูปที่ 4.18 Google apps script link

4. ส่วนของแพลตฟอร์ม Google apps script

การใช้งาน Web application นี้ ใช้สำหรับเรียกดูข้อมูลของโรคไข้เลือดออกออกมาเป็นกราฟ เพื่อให้แพทย์ได้วิเคราะห์และวินิจฉัยโครงสร้างของ RNA ของสายพันธุ์ไวรัสของโรคไข้เลือดออกว่าจะเกิดการกลายพันธุ์ (mutation) ของพันธุกรรมหรือไม่

ขั้นตอนการใช้งาน Web application นี้ มีการใช้งาน ดังรูปที่ 4.19 และ 4.20 ดังนี้

4.1 เปิดเว็บไซต์ขึ้นมาตามลิ้งนี้

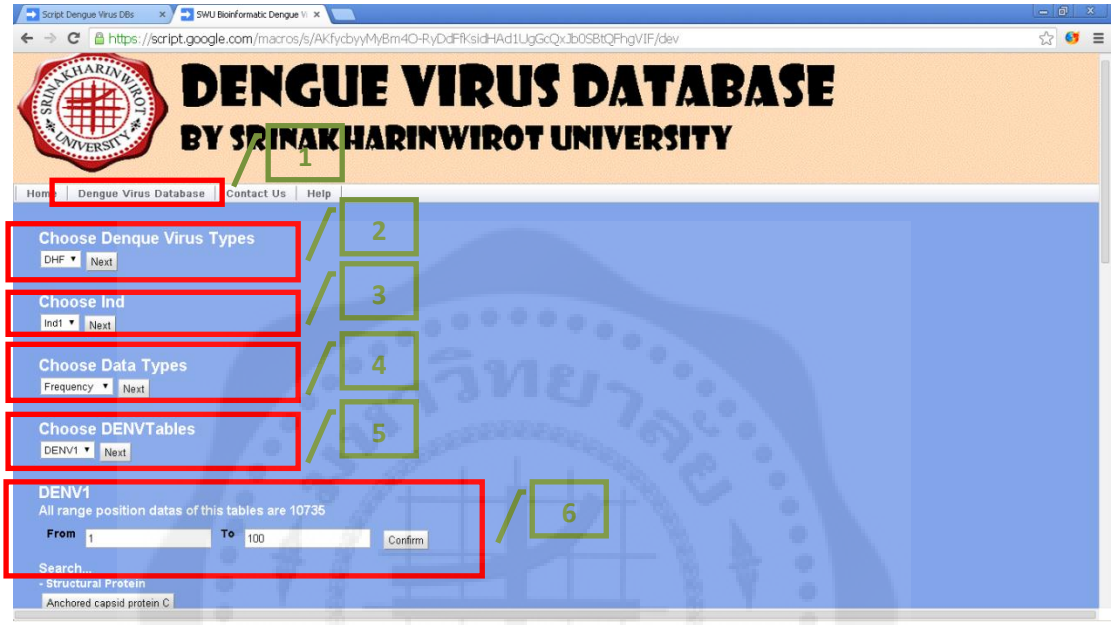
https://script.google.com/macros/s/AKfycbxLOKVsCRqklbPC2syGAUhiQBY4G3x_-zfCCTcPUskJIEVQQVwg/exec เลือกแถบเมนูที่ชื่อ Dengue virus database

4.2 เลือกชนิดของโรคไข้เลือกออก (Dengue Virus) ที่เป็นแบบ DF (Dengue Fever) หรือ DHF (Dengue hemorrhagic fever) แล้วคลิกปุ่ม next

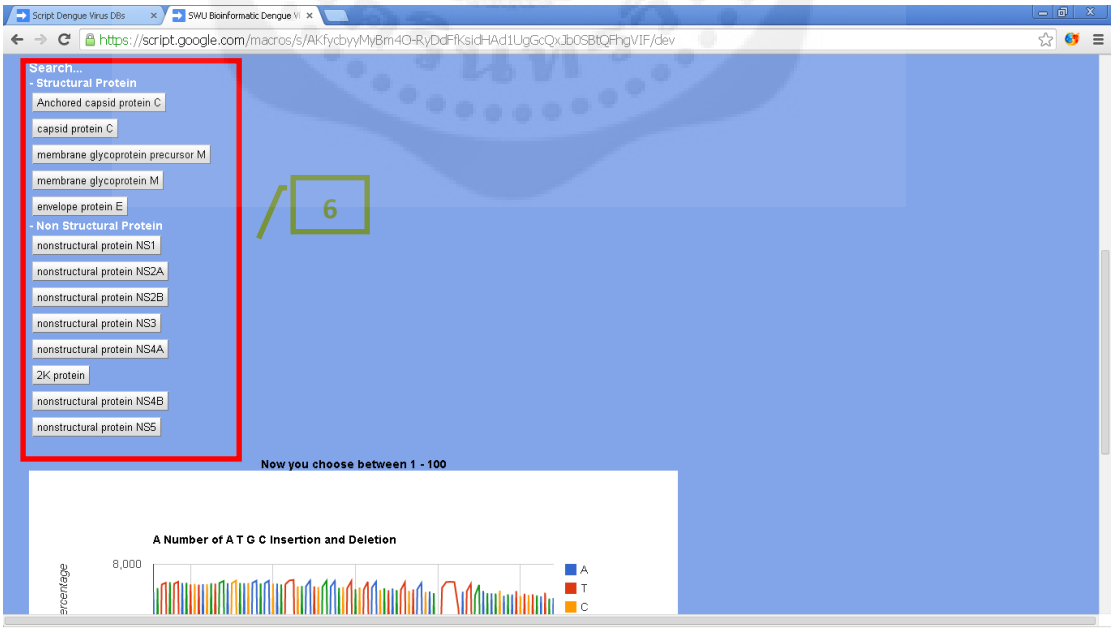
4.3 เลือกคนไข้ที่ต้องการ แทนด้วยคำว่า Ind ตามด้วยตัวเลข แล้วคลิกปุ่ม next

4.4 เลือกลักษณะการติดเชื้อ เรียกเป็น DENV1 ถึง DENV4 แล้วคลิกปุ่ม next

4.5 ใส่ค่าขอบเขตของตำแหน่งรหัสพันธุกรรมลงไป แล้วคลิก Confirm เพื่อทำการสร้างกราฟ หรือเลือกคลิกตามปุ่มของลักษณะโครงสร้างโปรตีนทั้งหมด 13 ชนิด ก็จะสร้างกราฟตามขอบเขตของโปรตีนแต่ละชนิดเช่นกัน

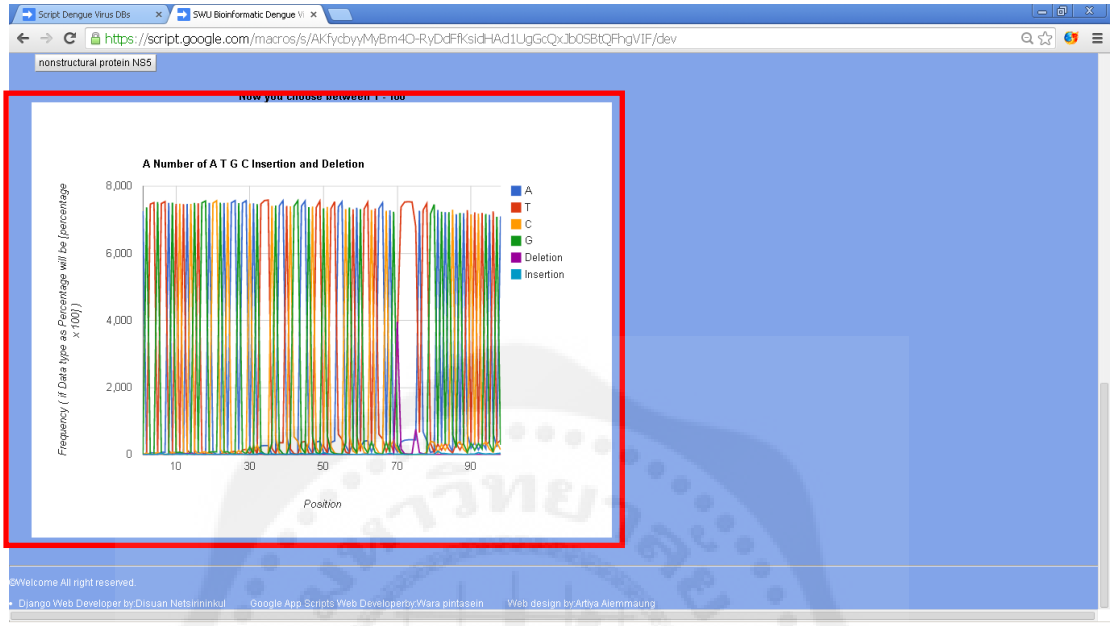


รูปที่ 4.19 วิธีใช้เว็บแอพพลิเคชันฐานข้อมูล Dengue virus โดย Google apps script



รูปที่ 4.20 วิธีใช้เว็บแอพพลิเคชันฐานข้อมูล Dengue virus โดย Google apps script

4.6 กราฟที่ได้มาจากการใส่ค่าขอบเขต 1-100 ดังรูปที่ 4.21



รูปที่ 4.21 การแสดงข้อมูลกราฟของ Google apps script

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

1. สรุปผลการดำเนินงาน

โครงงานนี้เป็นการจัดทำเว็บแอพพลิเคชันฐานข้อมูลซ้ำซ้อนเชื้อไวรัสไข้เลือดออกเพื่อนำไปใช้งานจริงโดยผู้ที่เชี่ยวชาญในสาขาที่เกี่ยวข้อง และผู้ที่กำลังศึกษาที่มาความสนใจในเรื่องเชื้อไวรัสไข้เลือดออก โดยเลือกใช้แพลตฟอร์ม Django และ Google apps script สร้างเว็บแอพพลิเคชันเพื่อเปรียบเทียบเพื่อหาจุดบกพร่องข้อได้เปรียบเสียเปรียบของแต่ละแพลตฟอร์ม

เนื่องจากในปัจจุบันมีหลากหลายแพลตฟอร์มที่เป็นที่นิยมให้เลือกใช้ ซึ่งแต่ละแพลตฟอร์มมีภาษาเขียนที่ต่างกัน ส่วนที่น่าสนใจคือ แพลตฟอร์มใดที่ใช้งานง่าย รวดเร็ว มีระเบียบ สามารถปรับปรุงแก้ไขได้ง่าย และทำงานอย่างมีประสิทธิภาพ

จากการทำโครงงานปริญญานิพนธ์นี้ขึ้นมา ทำให้ได้รับความรู้และเข้าใจในเรื่องต่างๆที่เกี่ยวข้องกับโครงงานตลอดจนได้พบกับปัญหา ซึ่งต้องแก้ไขปัญหาต่างๆบางปัญหาอาจใช้ระยะเวลามากเนื่องจากต้องค้นหาหรือสอบถามคำแนะนำจากผู้มีประสบการณ์ ซึ่งจากการแก้ไขปัญหาที่ผ่านมาทำให้เราได้รับความรู้และประสบการณ์ใหม่ๆเพิ่มมากขึ้นจากการค้นคว้า และได้เรียนรู้การสร้างเว็บแอพพลิเคชันโดย Django ที่ใช้ภาษาไพธอน และ Google apps script ที่ใช้ภาษาที่มีความคล้ายคลึงกับภาษาจาวาสคริปท์ เป็นภาษาเขียนโปรแกรม อีกทั้งวิธีการใช้งาน MySQL ในการจัดการฐานข้อมูล HTML5 และ CSS3 ในการตกแต่งหน้าเว็บไซต์

สำหรับการทำโครงงานปริญญานิพนธ์ฉบับนี้ สามารถสรุปได้ดังนี้ คือ เว็บแอพพลิเคชันที่สร้างขึ้นมานั้นสามารถใช้งานได้ง่าย มีความรวดเร็วในการประมวลผล เรียกดูข้อมูลและแสดงกราฟได้ตามที่ต้องการ ส่วนในด้านการเปรียบเทียบระหว่าง Google apps script และ Django นั้นพบว่าในด้านความสามารถของทั้งสองแพลตฟอร์มมีความสามารถในการทำงานที่แตกต่างกันเนื่องจากมีภาษาเขียนและรูปแบบการทำงานที่แตกต่างกัน โดยในเรื่องการปรับปรุงแก้ไขข้อมูล Django สามารถเข้าไปแก้ไขข้อมูลได้ง่ายกว่าเนื่องจาก Django เป็นแพลตฟอร์มที่ทำงานในรูปแบบ MVC (Model View Controller) ทำให้ข้อมูลถูกจัดเรียงอย่างเป็นระเบียบอีกทั้งยังสามารถร้องรับข้อมูลขนาดใหญ่ได้ ส่วน Google apps script จะได้เปรียบทางด้านความรวดเร็วในการเขียนโปรแกรม

เนื่องจากเขียนอยู่บนหน้าใดหน้าหนึ่งเท่านั้น ด้วยเหตุนี้จึงทำให้สามารถแก้ไขข้อมูลได้รวดเร็วมากกว่า Django

2. ปัญหาการดำเนินงาน

จากการศึกษาและทำการทดลองในโครงงาน พบปัญหาดังนี้

2.1 Django จัดว่าเป็นแพลตฟอร์มที่ยังไม่เป็นที่นิยมอย่างแพร่หลายทั้งในและต่างประเทศ ทำให้การศึกษาก่อนเริ่มทำโครงงานเป็นเป็นอย่างล่าช้า เนื่องจากต้องรวบรวมข้อมูลเป็นเวลานาน

2.2 เนื่องจากเครื่องเซิร์ฟเวอร์มีปัญหาทำให้ไม่สามารถ Deploy โปรเจค Django ขึ้นบนเซิร์ฟเวอร์ได้

2.3 Google apps script เป็นแพลตฟอร์มที่ต้องเขียนโปรแกรมบน Spreadsheet ของ Google ทำให้ต้องใช้ Internet ในการเขียนโปรแกรม ถ้าหาก Internet ล่มหรือช้า Google apps script จะไม่สามารถดำเนินงานต่อได้

3. ข้อเสนอแนะสำหรับการพัฒนาโครงงาน

3.1 โครงงานนี้ทั้งสองแพลตฟอร์มยังสามารถเพิ่มฐานข้อมูลของการติดเชื้อไวรัสโรคไข้เลือดออกเข้าไปอีกได้เป็นจำนวนมาก

3.2 ในส่วนของ Django แพลตฟอร์ม ควรทำหน้าเว็บให้แสดงผลออกมาในรูปแบบ Ajax

3.3 ในส่วนของ Google app script ในการแสดงผลกราฟ ควรทำการเซฟไฟล์เป็นรูปกราฟสกุล .png และไฟล์ฐานข้อมูล .csv ได้เหมือนกับการแสดงผลกราฟที่สร้างขึ้นโดย Django

3.4 ในส่วนของ Google app script สามารถเพิ่มเติมลูกเล่นการใช้งานเว็บแอพพลิเคชันได้อีกมากมาย เช่น สามารถเพิ่มการเรียกดูข้อมูลบนกราฟโดยการเพิ่มปุ่มเรียกดูข้อมูลที่แสดงแต่ข้อมูลที่เราต้องการให้ปรากฏบนกราฟได้

เอกสารอ้างอิง

Django Software Foundation. (2005-2014). **Django documentation.** Retrieved February 23, 2014, from https://docs.djangoproject.com/en/1.6/intro/tutorial01/

Refsnes Data. (1999-2014). **HTML Tutorial.** Retrieved February 23, 2014, from http://www.w3schools.com/html/html5_intro.asp

FeedBurner. (2010, January 1). **MySQL Commands.** Retrieved February 23, 2014, from http://www.pantz.org/software/mysql/mysqlcommands.html

Python Software Foundation. (2001-2014). **The Python Tutorial.** Retrieved February 23, 2014, from http://docs.python.org/2/tutorial/index.html

Google Developers. (2014, March 19). **Google apps script.** Retrieved February 23, 2014, from https://developers.google.com/apps-script/

Bootstrap members. (2011-2014). **Bootstrap CSS3.** Retrieved February 23, 2014, from http://getbootstrap.com/getting-started/

ภาคผนวก

ชื่อและรหัสการเข้าใช้ Google app script

สามารถเข้าไปดู Code โปรแกรม ได้โดย

1. เข้าใช้งาน Google account ดังต่อไปนี้

Username : **dengueDBSwu2013**

Password : **eetelswu2013**

2. เข้า Google drive และเข้าไฟล์ Script ที่ชื่อว่า Script Dengue Virus DBs หรือ https://script.google.com/d/1ZujOOXipoUQQDhTpVklBuZ4Uan_QkwZ81qN4chVEhQfTpvVD8wV2o49D/edit?usp=drive_web

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ - สกุล	นายดิศวร เนตรศิรินิลกุล
วัน เดือน ปีเกิด	21 กันยายน 2534
สถานที่เกิด	ลำปาง
สถานที่อยู่ปัจจุบัน	135/10 ถ.พหลโยธิน ต.สวนดอก อ.เมือง
	จ.ลำปาง 52100
โทรศัพท์	086-915-5947
Email	dn10tenz@gmail.com

ประวัติการศึกษา

พ.ศ. 2547 – 2549 มัธยมต้น โรงเรียนลำปางกัลยาณี จังหวัดลำปาง

พ.ศ. 2550 – 2552 มัธยมต้น โรงเรียนลำปางกัลยาณี จังหวัดลำปาง

พ.ศ. 2553 – 2556 ปริญญาตรี คณะวิศวกรรมศาสตร์ ภาควิศวกรรมไฟฟ้า แขนงโทรคมนาคม
มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ – สกุล	นายวรา ปินตาเสน
วัน เดือน ปีเกิด	16 เมษายน 2534
สถานที่เกิด	เชียงใหม่
สถานที่อยู่ปัจจุบัน	145 หมู่ 12 ต.บ้านโฮ่ง อ.บ้านโฮ่ง จ.ลำพูน 51130
โทรศัพท์	086-151-3765
Email	warapintasein@gmail.com

ประวัติการศึกษา

พ.ศ. 2547 – 2549 มัธยมต้น โรงเรียนมงฟอร์ตวิทยาลัย จังหวัดเชียงใหม่

พ.ศ. 2550 – 2552 มัธยมปลาย โรงเรียนมงฟอร์ตวิทยาลัย จังหวัดเชียงใหม่

พ.ศ. 2553 – 2556 ปริญญาตรี คณะวิศวกรรมศาสตร์ ภาควิศวกรรมไฟฟ้า แขนงโทรคมนาคม มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ – สกุล	นายอาติยะ เอี่ยมเมือง
วัน เดือน ปีเกิด	25 ธันวาคม 2534
สถานที่เกิด	กรุงเทพฯ
สถานที่อยู่ปัจจุบัน	1/1 หมู่ 6 ตำบล หนองสามวัง อำเภอ หนองเสือ จังหวัด ปทุมธานี 12170
โทรศัพท์	08-2578-2818
E-mail	Artie-smart@hotmail.com

ประวัติการศึกษา

พ.ศ. 2547-2549 มัธยมต้น โรงเรียนหนองเสือวิทยาคม จังหวัดปทุมธานี

พ.ศ. 2550-2552 มัธยมปลาย โรงเรียนหนองเสือวิทยาคม จังหวัดปทุมธานี

พ.ศ. 2553-2556 ปริญญาตรี คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมไฟฟ้าแขนงโทรคมนาคม มหาวิทยาลัยศรีนครินทรวิโรฒ