

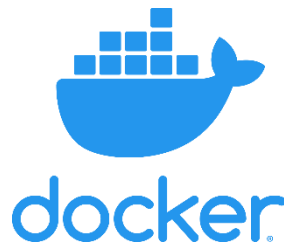
การดึงข้อมูลสถานการณ์การแพร่ระบาดโรคโควิด19 ด้วย Apache Airflow with Docker

1. Docker

Docker คือแพลตฟอร์มซอฟต์แวร์ที่ช่วยให้สร้าง ทดสอบ และติดตั้งแอปพลิเคชันให้ใช้จริงได้อย่างรวดเร็ว โดย Docker จะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server ด้วย Docker container ซึ่งจะมีสิ่งจำเป็นที่ซอฟต์แวร์ต้องใช้ในการเรียกใช้งาน ไลบรารี เครื่องมือสำหรับระบบ โค้ด และรันไทม์ นอกจากนี้จะมีส่วนที่เรียกว่า Docker Image ซึ่งภายในจะประกอบด้วย application ต่างๆ ที่มีการติดตั้งไว้เพื่อใช้งานสำหรับ service นั้นๆ รวมทั้งมีการ config ค่าต่างๆ ไว้เรียบร้อยแล้ว

การติดตั้ง Docker บนระบบปฏิบัติการ Window สามารถอ่านข้อกำหนดและขั้นตอนการติดตั้งโปรแกรมได้จาก

<https://docs.docker.com/desktop/windows/install/>



2. Apache airflow

Airflow เป็นแพลตฟอร์มที่ใช้ในการเขียนโปรแกรม กำหนดเวลาการทำงานและตรวจสอบการทำงาน workflow หรือ data pipeline ด้วยภาษา python โดย task ต่างๆ จะเขียนใน Directed Acyclic Graphs (DAGs) นอกจากนี้ user interface ยังทำให้เห็นการไหลของงาน (pipeline) ขณะทำงานได้ง่าย รวมถึงทำให้สามารถติดตามความคืบหน้าและสามารถแก้ไขเมื่อเกิดปัญหาได้ง่ายอีกด้วย

2.1 Running Airflow in Docker

2.1.1 Config Airflow Dockerfile เพื่อให้รู้ว่าต้องการ Image ไດ ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จาก docker hub

```
airflow > Dockerfile > ...
1 FROM ubuntu:20.04
2 RUN apt-get update
3 RUN apt-get install -y python3.8 python3-pip libmysqlclient-dev
4 RUN mkdir /opt/airflow
5 ENV AIRFLOW_HOME=/opt/airflow
6 ENV AIRFLOW__CORE__LOAD_EXAMPLES=False
7
8 RUN pip3 install apache-airflow==2.2.1 --constraint https://raw.githubusercontent.com/apache/airflow/constraints-2.2.1/constraints-3.8.txt
9 RUN pip3 install pandas beautifulsoup4 sklearn
10 RUN pip3 install apache-airflow-providers-mysql==2.1.1
11
12 RUN airflow db init
13 RUN airflow users create \
14     --username admin \
15     --password password \
16     --firstname Napat \
17     --lastname Sermsuwannasuk \
18     --role Admin \
19     --email 63606019@kmitl.ac.th
```

2.1.2. Config mysql Dockerfile เพื่อให้รู้ว่าต้องการเชื่อมต่อกับฐานข้อมูลใด

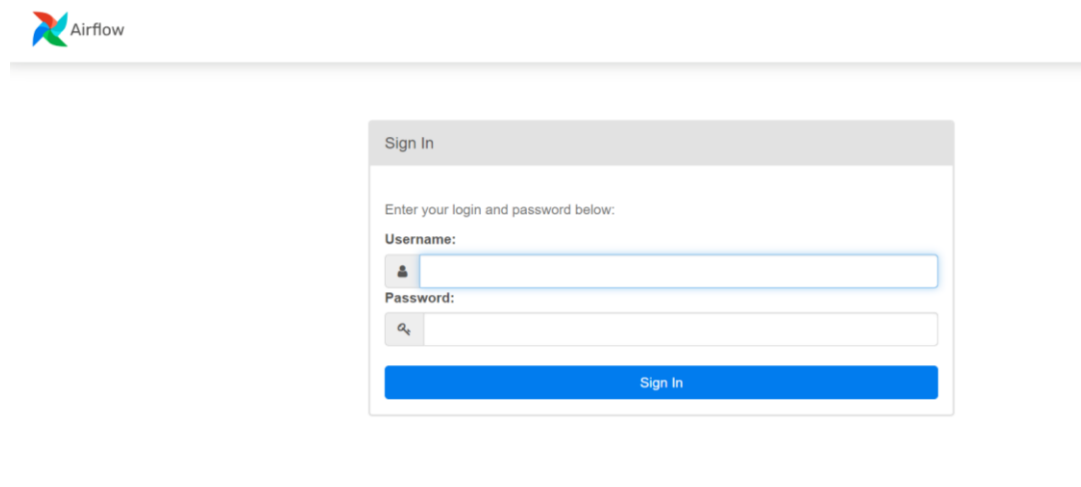
```
mysql > Dockerfile > ...  
1 FROM mysql/mysql-server:5.7  
2  
3 ENV MYSQL_ROOT_PASSWORD=password  
4 ENV MYSQL_DATABASE=testdb  
5 ENV MYSQL_USER=testdb  
6 ENV MYSQL_PASSWORD=testdb
```

2.1.3. Config docker-compose.yml เพื่อให้สามารถใช้งาน docker container ได้สะดวกขึ้น

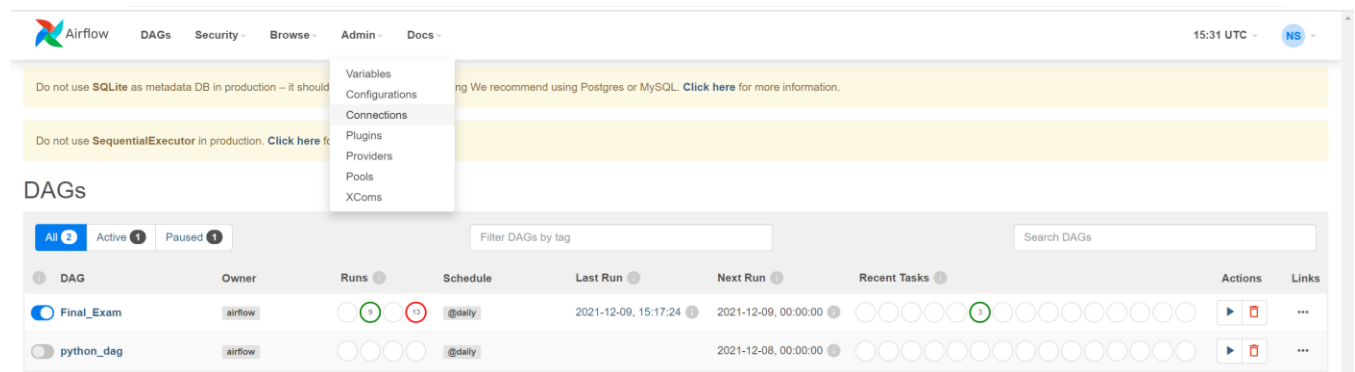
```
docker-compose.yml  
1 version: "3.9"  
2 services:  
3   airflow:  
4     build: ./airflow  
5     volumes:  
6       - ./dags:/opt/airflow/dags  
7       - ./logs:/opt/airflow/logs  
8       - ./plugins:/opt/airflow/plugins  
9     ports:  
10      - 8080:8080  
11     command: bash -c "airflow webserver --port 8080 & airflow scheduler"  
12  
13   mysql:  
14     #image: mysql/mysql-server:5.7  
15     build: ./mysql  
16     ports:  
17       - 3306:3306  
18  
19   phpmyadmin:  
20     image: phpmyadmin/phpmyadmin:5.1  
21     depends_on:  
22       - mysql  
23     restart: always  
24     ports:  
25       - '8088:80'  
26     environment:  
27       PMA_HOST: mysql  
28       PMA_PORT: 3306  
29
```

2.1.4 การสร้าง connection ระหว่าง airflow กับ database

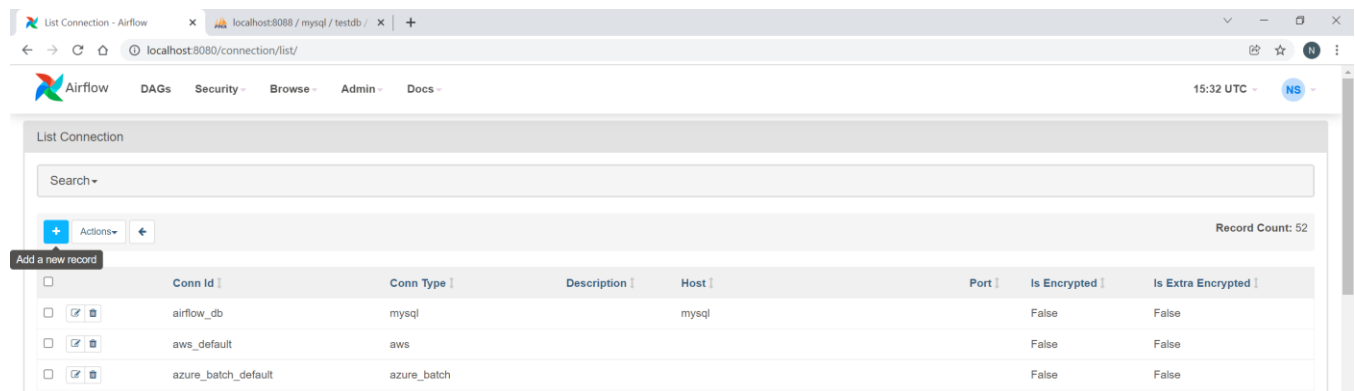
หลังจากขั้นตอน 2.1.3 ให้พิมพ์ docker-compose up เพื่อเริ่มต้นใช้งาน แล้วไปที่ <http://localhost:8080/> ทำการ sign-in ตามที่กำหนดไว้ใน Airflow Dockerfile



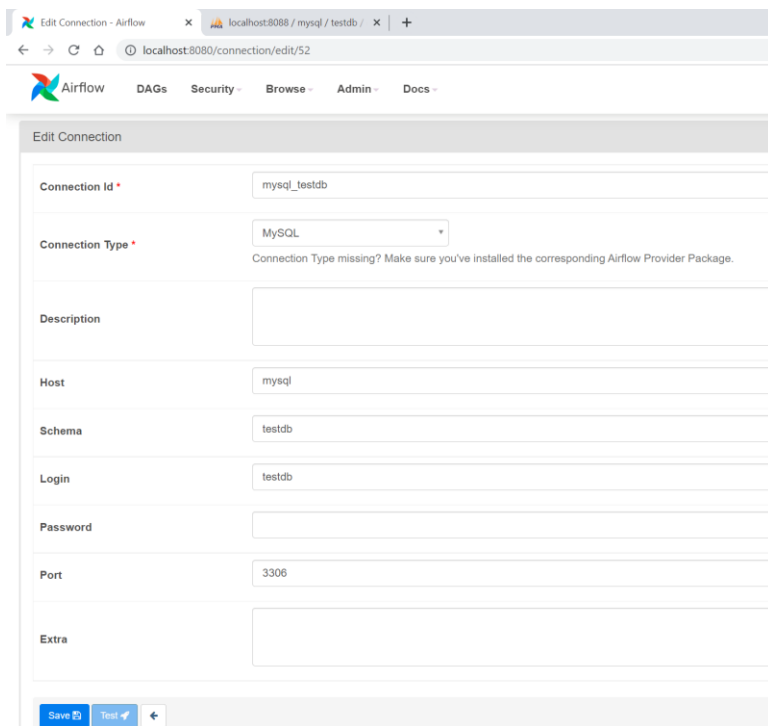
เมื่อเข้ามาแล้วให้เลือกที่ Admin >> Connection



กดเครื่องหมาย + เพื่อทำการสร้าง connection



ระบุข้อมูลตามที่กำหนดไว้ใน mysql Dockerfile และ docker-compose.yml จากนั้นกด save

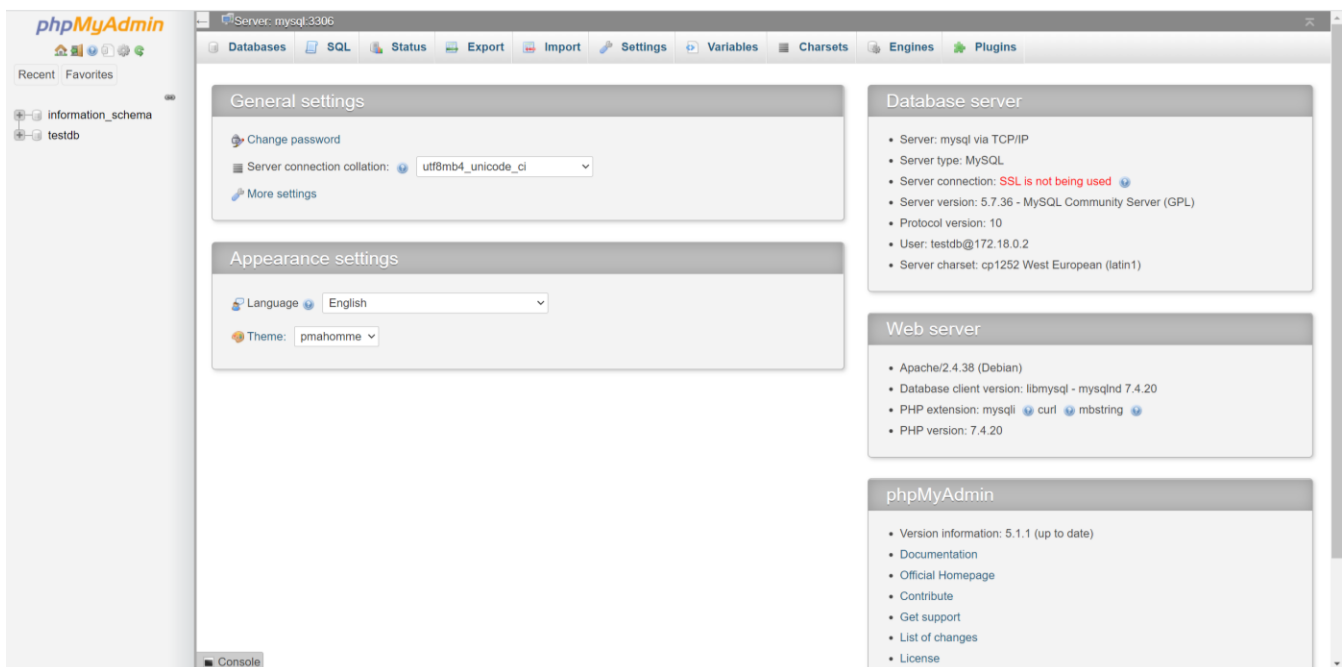


The screenshot shows the 'Edit Connection' form in the Airflow web interface. The form is titled 'Edit Connection' and contains the following fields:

- Connection Id ***: mysql_testdb
- Connection Type ***: MySQL (selected from a dropdown menu). Below this field, a message reads: 'Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.'
- Description**: (Empty text area)
- Host**: mysql
- Schema**: testdb
- Login**: testdb
- Password**: (Empty password field)
- Port**: 3306
- Extra**: (Empty text area)

At the bottom of the form, there are three buttons: 'Save' (with a disk icon), 'Test' (with a play icon), and a back arrow.

ฐานข้อมูลสามารถเข้าถึงได้จาก <http://localhost:8088/> ซึ่งในรายงานนี้กำหนดชื่อว่า “testdb”



3. แหล่งที่มาของข้อมูล

เป็นข้อมูลเกี่ยวกับสถานการณ์ผู้ติดเชื้อ COVID-19 อัปเดตรายวันผ่านเว็บไซต์ <https://covid19.ddc.moph.go.th/api> ซึ่งถูกจัดทำโดยศูนย์สารสนเทศ กรมควบคุมโรค โดยข้อมูลที่เราเลือกเก็บนี้เป็นข้อมูลรายงานสถานการณ์ COVID-19 ระลอก 3 (ตั้งแต่ 01/04/2021 -ปัจจุบัน) ซึ่งมีรายละเอียดดังตารางที่ 1

ตารางที่ 1 รายละเอียดข้อมูลรายงานสถานการณ์ COVID-19 ระลอก 3 (ตั้งแต่ 01/04/2021 -ปัจจุบัน)

Column Name	Data Type	comment
txn_date	date	วันแถลง
new_case	Int	จำนวนผู้ป่วยรายใหม่
total_case	Int	จำนวนผู้ป่วยสะสม
new_case_excludeabroad	Int	จำนวนผู้ป่วยรายใหม่ (ไม่นับมาจากต่างประเทศ)
total_case_excludeabroad	Int	จำนวนผู้ป่วยสะสม (ไม่นับมาจากต่างประเทศ)
new_death	Int	จำนวนผู้ป่วยตายรายใหม่
total_death	Int	จำนวนผู้ป่วยตายสะสม
new_recovered	Int	จำนวนผู้ป่วยรักษาหายรายใหม่
total_recovered	Int	จำนวนผู้ป่วยรักษาหายสะสม

4. Task ใน DAG

กำหนดวันที่เริ่มต้นคือวันที่ 1 เมษายน พ.ศ. 2564 และกำหนดเวลาทำงานเป็นทุกวัน โดยจะมีการแบ่งเป็น 3 task

```
default_args = {
    'start_date': datetime(2021, 4, 1)
}
@dag('Final_Exam', schedule_interval='@daily', default_args=default_args,
catchup=False)
```

4.1 extract เป็นส่วนที่ทำการดึงข้อมูล API จากแหล่งข้อมูลที่ต้องการ และคืนค่ากลับให้อยู่ในรูปแบบ json

```
def extract():
    import pandas as pd
    import requests
    import json
    target_url = 'https://covid19.ddc.moph.go.th/api/Cases/timeline-cases-all'
    raw_json = requests.get(target_url).text
    data = json.loads(raw_json)
    df = pd.DataFrame(data)
    #print(df.describe())
    return df.to_json()
```

4.2 transform ทำการคำนวณอัตราการรักษาหายต่อวันและอัตราผู้ป่วยรายใหม่ต่อวัน แล้วเก็บข้อมูลในรูปแบบ json

```
def transform(val):
    import pandas as pd
    import json
    df1 = pd.read_json(val)
    a = round(((df1.new_recovered/df1.total_case)*100),2)
    b = round(((df1.new_case/df1.total_case)*100),2)
    d = {'Date': df1.txn_date , 'recovered': a, 'new_case': b}
    df2 = pd.DataFrame(d)
    return df2.to_json()
```

4.3 load_to_mysql ทำการสร้างตารางชื่อ covid ด้วยคำสั่ง c.execute (หากยังไม่มีการสร้างตารางนี้) และทำการวนลูปเพื่อเก็บข้อมูลที่ละแถวเข้า database ด้วย SQL

```
def load_to_mysql(val):
    import pandas as pd
    import json
    hook = MySQLHook(mysql_conn_id='mysql_testdb')
    conn = hook.get_conn()
    c = conn.cursor()
    df3 = pd.read_json(val)

    c.execute('''
CREATE TABLE IF NOT EXISTS covid
(YYYY_MM_DD DATE NOT NULL,
recovered_percent decimal(4,2) NOT NULL,
new_case_percent decimal(4,2) NOT NULL);''')

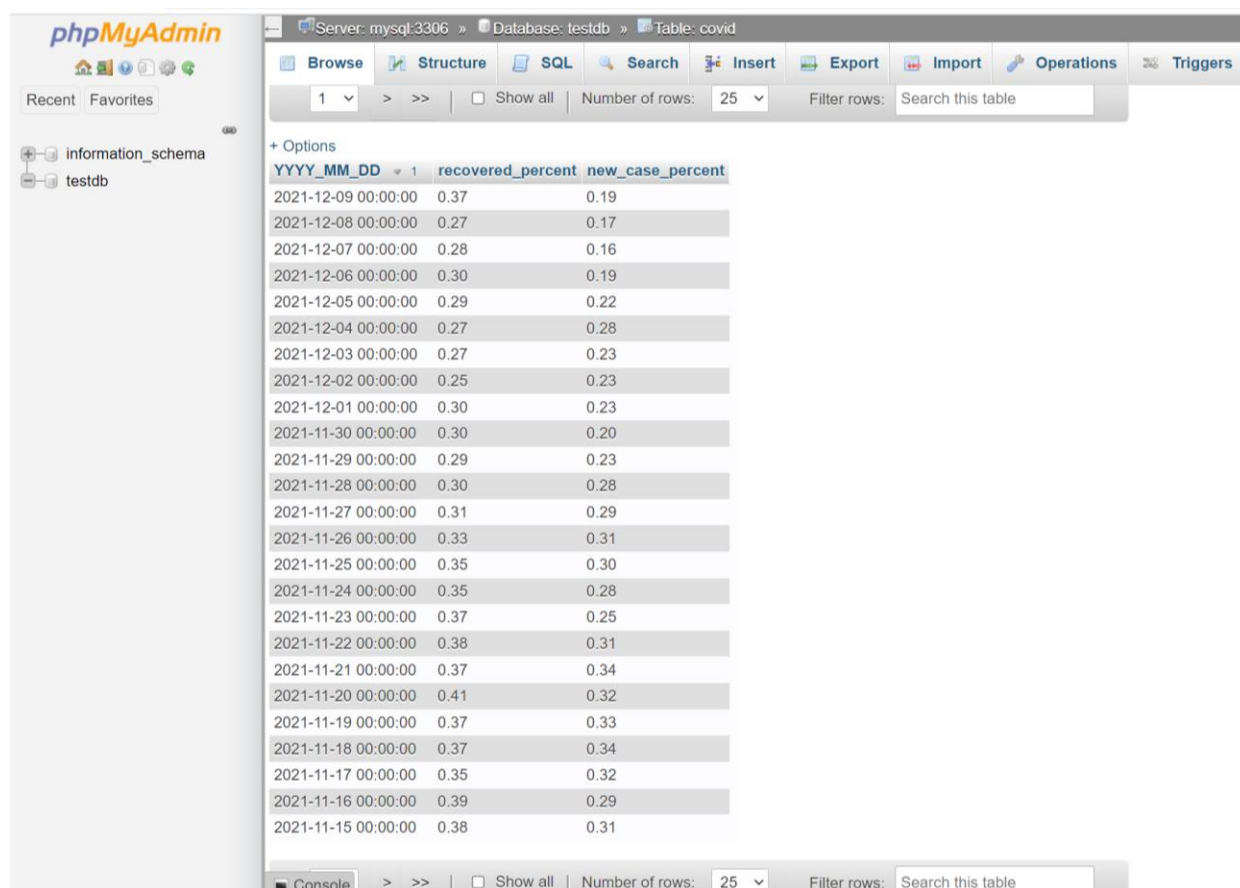
    for index, row in df3.iterrows():
        query = """
INSERT INTO covid (YYYY_MM_DD, recovered_percent, new_case_percent)
VALUES ('{Date}',{recovered},{new_case})
""".format(Date = row['Date'], recovered = row['recovered'],
            new_case = row['new_case'])

        hook.run(sql=query)
```

5. ข้อมูลที่เก็บใน database

ตารางที่ 2 ประเภทข้อมูลที่เก็บลง database

Column Name	Data Type	comment
YYYY_MM_DD	date	วันแถลง
recovered_percent	decimal(4,2)	อัตราการรักษาหายต่อวัน
new_case_percent	decimal(4,2)	อัตราผู้ป่วยรายใหม่ต่อวัน



YYYY_MM_DD	recovered_percent	new_case_percent
2021-12-09 00:00:00	0.37	0.19
2021-12-08 00:00:00	0.27	0.17
2021-12-07 00:00:00	0.28	0.16
2021-12-06 00:00:00	0.30	0.19
2021-12-05 00:00:00	0.29	0.22
2021-12-04 00:00:00	0.27	0.28
2021-12-03 00:00:00	0.27	0.23
2021-12-02 00:00:00	0.25	0.23
2021-12-01 00:00:00	0.30	0.23
2021-11-30 00:00:00	0.30	0.20
2021-11-29 00:00:00	0.29	0.23
2021-11-28 00:00:00	0.30	0.28
2021-11-27 00:00:00	0.31	0.29
2021-11-26 00:00:00	0.33	0.31
2021-11-25 00:00:00	0.35	0.30
2021-11-24 00:00:00	0.35	0.28
2021-11-23 00:00:00	0.37	0.25
2021-11-22 00:00:00	0.38	0.31
2021-11-21 00:00:00	0.37	0.34
2021-11-20 00:00:00	0.41	0.32
2021-11-19 00:00:00	0.37	0.33
2021-11-18 00:00:00	0.37	0.34
2021-11-17 00:00:00	0.35	0.32
2021-11-16 00:00:00	0.39	0.29
2021-11-15 00:00:00	0.38	0.31

6. ประโยชน์ที่ได้รับ

สามารถเก็บข้อมูลเกี่ยวกับสถานการณ์การแพร่ระบาดของโรคโควิด19 แบบรายวันโดยอัตโนมัติ ทำให้ไม่ต้องใช้เวลาทำการเข้าเว็บไซต์เพื่อดึงข้อมูลด้วยตนเองทุกวัน รวมถึงยังสามารถเก็บข้อมูลตามที่ต้องการหรือสนใจใน database ได้เอง โดยในงานนี้กำหนดให้คำนวณอัตราการรักษาหายต่อวันและอัตราผู้ป่วยรายใหม่ต่อวันแล้วจัดเก็บลง database ซึ่งจากข้อมูลที่ทำกรเก็บรวบรวมนี้จะทำให้ทราบถึงแนวโน้มการรักษาและผู้ติดเชื้อรายใหม่ต่อวันในประเทศไทย ซึ่งจะทำให้สามารถวิเคราะห์และปรับเปลี่ยนมาตรการป้องกันหรือมาตรการรับมือต่อสถานการณ์ได้อย่างเหมาะสม