

**อีลาสติกเสิร์ช**

**Elasticsearch**

**นายบุรินทร์ พันธ์ชาติ**

**Burin Panchat**

**รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรี**

**หลักสูตรวิศวกรรมซอฟต์แวร์**

**มหาวิทยาลัยวลัยลักษณ์**

**A Report Submitted in Partial Fulfillment of the Requirements**

**for the Degree of Software Engineering**

**Walailak University**

**ปีการศึกษา 2562**



**อีลาสติกเสิร์ช**

**Elasticsearch**

**นายบุรินทร์ พันธ์ชาติ**

**Burin Panchat**

**รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรี**

**หลักสูตรวิศวกรรมซอฟต์แวร์**

**มหาวิทยาลัยวลัยลักษณ์**

**A Report Submitted in Partial Fulfillment of the Requirements**

**for the Degree of Software Engineering**

**Walailak University**

**ปีการศึกษา 2562**

**บทคัดย่อ**

elasticsearch เป็นเครื่องมือที่สามารถนำมาประยุกค์ใช้ในการสร้าง search engine ได้ แต่การ search ธรรมดาก็สามารถให้ได้เหมือนกัน จึงเกิดความคิดที่ว่า elasticsearch ค้นหาได้ดีกว่าการทำ search ทั้วไปยังไง

ผู้จัดทำรายงานอยากที่จะทราบถึงการเปรียบเทียบการ search ด้วยวิธีต่างๆเทียบกับการสร้าง search ด้วย elasticsearch ว่ามีความแตกต่างกันอย่างไรมากน้อยเพียงใด เพื่อวัดประสิทธิภาพในการทำงานต่างๆของการ search

**คำสำคัญ :** Search engine**,** Elasticsearch**,** Logstash**,** Kibana

**กิตติกรรมประกาศ (Acknowledgment)**

รายงานฉบับนี้เสร็จสมบูรณ์ได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์ความกรุณาเป็นอย่างดีจาก ผู้ช่วยศาสตราจารย์ ดร.พุทธิพร ธนธรรมเมธี อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษาแนะนำและให้ข้อเสนอแนะต่างๆตลอดจนตรวจสอบ แก้ไขปรับปรุงข้อบกพร่องต่างๆด้วยความเอาใจใส่อย่างดียิ่ง จนทำให้รายงานฉบับนี้สำเร็จได้ด้วยดี ซึ่งผู้จัดทำรายงานรู้สึกซาบซึ้งในความกรุณา จึงขอกราบขอบพระคุณเป็นอย่างมากไว้ ณ โอกาสนี้

กราบขอบพระคุณคณาจารย์หลักสูตรวิศวกรรมซอฟต์แวร์ สำนักสารสนเทศศาสตร์ ที่ช่วยให้คำแนะนำในการศึกษา ซึ่งเป็นประโยชน์ต่อการศึกษาและการค้นหาของผู้วิจัย และเพื่อนๆพี่ๆหลักสูตรวิศวกรรมซอฟต์แวร์

บุรินทร์ พันธ์ชาติ

**สารบัญ**

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3ประโยชน์ที่คาดว่าจะได้รับ 1

1.4 เทคโนโลยีที่ทำการศึกษา 1

บทที่ 2 ทฤษฏีและเครื่องมือที่เกี่ยวของ 2

2.1 เทคโนโลยีที่ศึกษา 2

2.1.1 Elasticsearch 2

2.1.2 Kibana 4

2.1.3 Logstash 4

2.2 ทฤษฏีที่เกี่ยวข้อง 5

2.2.1 TF-IDF 5

2.2.2 RESTful API 5

2.2.3 SPA (Single Page Application) 6

2.3 งานวิจัยหรือระบบงานที่ใกล้เคียง 7

บทที่ 3 การดำเนินงานและผล 9

3.1 ขอบเขตการดำเนินงาน 9

3.1.1 คุณลักษณะของระบบ 9

3.1.2 ฟังก์ชันงานของระบบ 9

3.2 ขั้นตอนการพัฒนา 9

3.2.1 ขั้นตอนเตรียมการพัฒนา 9

3.2.2 ขั้นตอนการพัฒนา 11

3.2.3ขั้นตอนการทดสอบ 17

3.3 ผลการดำเนินงาน 17

บทที่ 4 อภิปรายและสรูปผลการดำเนินงาน 18

เอกสารอ้างอิง 19

**สารบัญภาพ**

ภาพที่ 2.1 Logo Elasticsearch 2

ภาพที่ 2.2 ภาพตัวอย่างการอธิบายการทำ index 3

ภาพที่ 2.3 Logo Kibana 4

ภาพที่ 2.4 Logo Logstash 4

ภาพที่ 2.5 RESTful API 5

ภาพที่ 3.1 เช็ค version JDK 9

ภาพที่ 3.2 ดาวโหลด Elasticsearch 10

ภาพที่ 3.3 ดาวโหลด Kibana 10

ภาพที่ 3.4 ดาวโหลด Logstash 11

ภาพที่ 3.5 ภาพข้อมูล Elasticsearch 11

ภาพที่ 3.6 ภาพข้อมูล MongoDB 12

ภาพที่ 3.7 ภาพตัวอย่างเว็บไซต์ 12

ภาพที่ 3.8 การดาวโหลดComposer 13

ภาพที่ 3.9 การติดตั้ง Composer 13

ภาพที่ 3.10 ข้อมูล 14

ภาพที่ 3.11 หน้าจออินพุต 14

ภาพที่ 3.12 หน้าจอเอาท์พุต 15

ภาพที่ 3.13 การconnect php กับ Elasticsearch 15

ภาพที่ 3.14 Logstashs.conf 16

ภาพที่ 3.15 การค้นหาที่พักNYC 17

# บทที่ 1

**บทนำ**

## 1.1 ความสำคัญและที่มาของปัญหา

แสดงเห็นและผล ความจำเป็น หรือความน่าสนใจของการศึกษาในครั้งนี้

## 1.2 วัตถุประสงค์

1. เพื่อได้ศึกษาแนวคิด Elasticsearch
2. เพื่อได้เรียนรู้การนำ Elasticsearch มาใช้ในการช่วยพัฒนาเว็บไซต์
3. เพื่อใช้ความรู้นี้ในสร้าง search engine

## 1.3ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ทักษะในการพัฒนาที่จำเป็นในการทำงานในอนาคตจากการใช้ Elasticsearch
2. ได้พัฒนาทักษะความสามารถของตนเองในการเขียนโปรแกรม และการค้นหาความรู้ใหม่ๆ จากเครื่องมือที่ไม่เคยใช้งานและรู้จักมาก่อน
3. ได้เตรียมตัวในการออกไปปฏิบัติสหกิจศึกษา

## 1.4 เทคโนโลยีที่ทำการศึกษา

* Elasticsearch Version : 7.6.1
* Kibana Version : 7.6.1
* Logstash Version : 7.6.1

# บทที่ 2

**ทฤษฏีและเครื่องมือที่เกี่ยวข้อง**

## 2.1 เทคโนโลยีที่ศึกษา

### 2.1.1 Elasticsearch



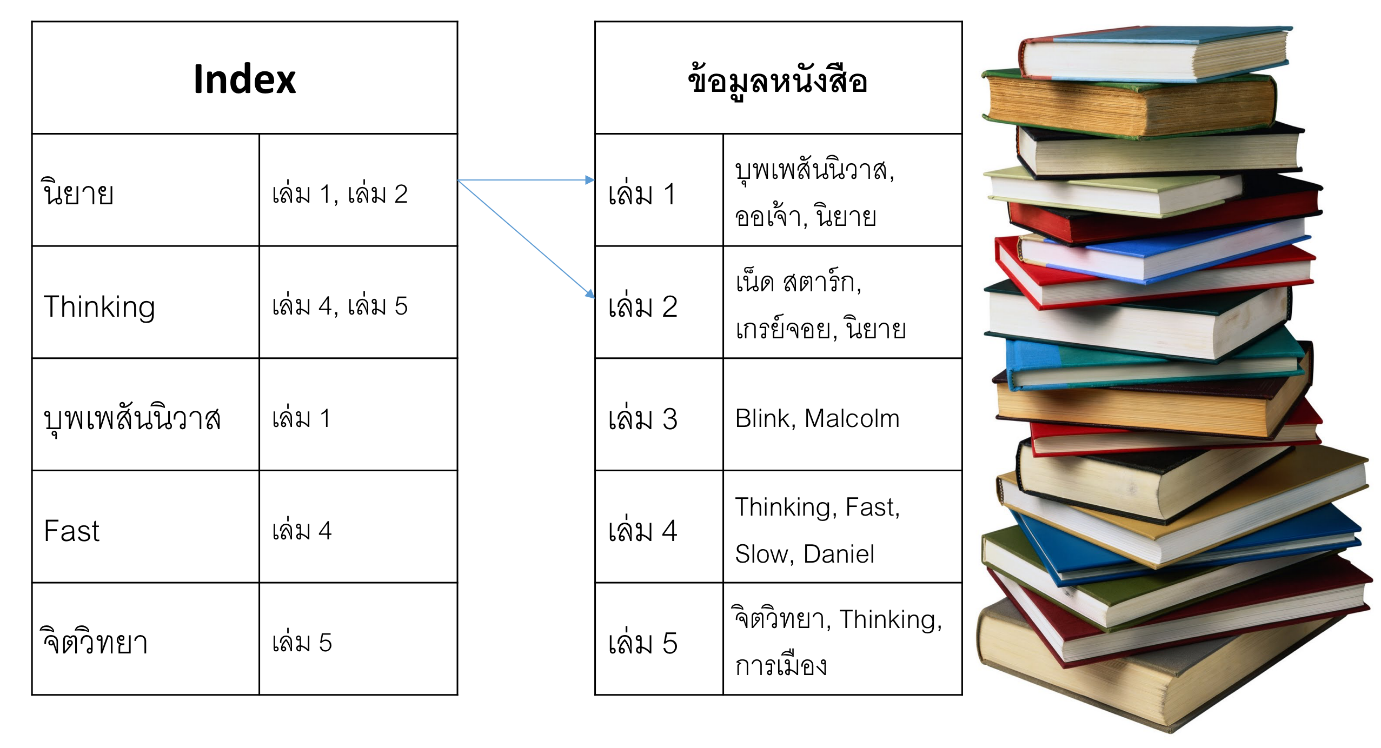
#### ภาพที่ 2.1 Logo Elasticsearch

https://www.seekpng.com/png/detail/221-2211816\_elk-stack-logo.png

Elasticsearch คือ ที่เก็บข้อมูลแบบกระจาย(distributed data store) ตัวหนึ่งที่มีความสามารถหลักๆ คือ search (พัฒนาต่อยอดมาจาก Lucene) ที่ไม่ใช่ search แบบธรรมดาแต่เป็นการกระจายการค้นหา (distributed search) ที่ทุกๆ ฟิลด์ข้อมูลที่เก็บจะถูกทำ index ไว้ ทำให้ความเร็วในการ search กับข้อมูลขนาดมหาศาลนั้นแทบจะเป็น real-time และการติดต่อกับส่วน data store สามารถทำได้โดยผ่าน RESTful API ในการอ่านเขียนข้อมูลกับตัว Elasticsearch

Search Engine ทุกตัวจะมีองค์ประกอบสามส่วนใหญ่ๆ ที่เหมือนกัน นั่นคือ การทำระบบ ดัชนีค้นหา (Index), ระบบคะแนนผลลัพธ์ (Relevancy) และระบบอำนวยความสะดวกผู้ใช้

หากไม่มีการทำดัชนีค้นหาขึ้นมาก่อน ผู้ใช้งานจะไม่สามารถค้นหาอะไรในเว็บของเราได้เลย ซึ่งระบบดัชนีของ Search Engine ทั้งหลายนั้นมีชื่อเรียกเฉพาะว่า Inverted Index ซึ่งมีวิธีทำงานดังนี้



#### ภาพที่ 2.2 ภาพตัวอย่างการอธิบายการทำ index

https://miro.medium.com/max/1400/1\*kI\_Pg1iXpQNBrtMHYJAl9g.png

สมมติทำระบบสำหรับค้นหาข้อมูลในหนังสือ ดังนั้น ข้อมูลดิบที่จะต้องเก็บในฐานข้อมูลก็คือ เนื้อหาตัวอักษรทั้งหมดในหนังสือ แต่ข้อมูลนี้ไม่สามารถนำไปใช้ค้นหาได้ (หากฝืนจะค้นหาคำบางคำภายใต้กองหนังสือเป็นพันเป็นหมื่นเล่มก็จะใช้เวลานาในการค้นหา)

ระบบจะทำการจัดเก็บระบบ Inverted Index โดยนำ แต่ละคำภายในหนังสือมาจัดเรียงใหม่โดยทำคล้ายๆ กับสารบัญเพื่อแจ้งว่าคำๆ นั้น บรรจุอยู่ในหนังสือเล่มใดบ้าง

เวลาผู้ใช้ค้นหาระบบ เช่นค้นหาคำว่า “นิยาย” ระบบ Search Engine จะเข้าไปค้นหาใน Inverted Index เพื่อดูว่า “นิยาย” อยู่ในหนังสือเล่มไหนบ้าง แล้วจึงส่งผลลัพธ์ออกไปให้กับผู้ใช้ว่าอยู่ใน เล่ม 1 และ 2 ตามตัวอย่างในรูป หากคำๆ นั้นมันปรากฎอยู่ในหนังสือหลายเล่ม เวลา Search Engine แสดงผลลัพธ์ จะเอาอะไรตัดสินว่า เล่มไหนต้องขึ้นอันดับ 1 เล่มไหนต้องอยู่อันดับสุดท้าย

Elasticsearch จะใช้วิธีคำนวนคะแนนที่มีชื่อว่า TF-IDF (Term Frequency-Inverse Document Frequency) โดยแต่ละตัวจะมีความหมายดังนี้

* Term Frequency: ยิ่งมีคำในเอกสารมากเท่าไหร่ คะแนนก็ยิ่งมากเท่านั้น
* Inverse Document Frequency: ยิ่งคำๆ นั้น ไปปรากฎบนเอกสารจำนวนน้อยเท่าไหร่ คะแนนก็ยิ่งมาก

### 2.1.2 Kibana



#### ภาพที่ 2.3 Logo Kibana

https://www.seekpng.com/png/detail/221-2211816\_elk-stack-logo.png

เครื่องมือ Visualize สำหรับแสดงผลข้อมูลจาก Elasticsearch ในรูปแบบต่างๆ เช่น กราฟแบบต่างๆ ตาราง แผนที่ และสามารถสร้างการแสดงผลข้อมูล หรือ Dashboard ได้ตามความต้องการ และยังเป็นเครื่องมีที่ช่วยให้การทำงานกับ Elasticsearch ได้ง่ายขึ้น หากไม่ใช้ Kibana ช่วย เวลาทำงานกับ Elasticsearch จะต้องทำผ่าน Command Line ซึ่ง Kibana นี้จะช่วยลดการเสียเวลาในการทำงานกับ Elasticsearch บน  
Command Line

### 2.1.3 Logstash



#### ภาพที่ 2.4 Logo Logstash

https://www.seekpng.com/png/detail/221-2211816\_elk-stack-logo.png

Logstash คือเครื่องมือในการรวบรวม และแปลงข้อมูล โดยจะประกอบไปด้วย Inputs คือการรับข้อมูลในรูปแบบต่าง และทำการ Filters วิเคราะห์ เปลี่ยนแปลงข้อมูล และสร้าง Outputs เพื่อส่งต่อข้อมูลไปยัง Elasticsearch ส่วนการทำงานของ Logstash ประกอบไปด้วย 4 ส่วน คือ

1. Input คือ ข้อมูลเข้าต่างๆ หรือ source data
2. Codec คือ การเข้ารหัสข้อมูล
3. Filter คือ การกรองข้อมูลให้อยู่ในรูปแบบที่ต้องการ
4. Output คือ ข้อมูลออกในรูปแบบต่างๆ หรือ ที่จัดเก็บผลลัพธ์ต่างๆ

## 2.2 ทฤษฏีที่เกี่ยวข้อง

### 2.2.1 TF-IDF

TF-IDF หรือ Term Frequency-Inverse Document Frequency — เป็นหนึ่งในวิธีหา “คำ(term)” ที่สำคัญ ใน “เอกสาร(document)” โดยดูจากเนื้อหาของเอกสารทั้งหมด มักจะใช้ในงานพวก Information-retrieval หรือ Text mining

TF-IDF เกิดจากผลคูณของสองค่าคือ TF(Term Frequency) กับ IDF(Inverse Document Frequency)

**TF (Term Frequency)**

ความถี่ของคำ(term) และเนื่องจากว่าแต่ละเอกสาร(document) อาจจะมีความยาวที่แตกต่างกัน ดังนั้นการคำนวณ Term Frequency จึงมักจะหารด้วยความยาวของ document (จำนวนคำทั้งหมดใน document นั้น) วิธีการคำนวณ Term Frequency มีหลายวิธี เช่น Raw counts, Log normalization, Double normalization เป็นต้น

**IDF (Inverse Document Frequency)**

ใช้สำหรับวัดความสำคัญของ คำ(term) ใน เอกสาร(document) ทั้งหมด หมายถึง ถ้า คำ (term) เจอในหลาย เอกสาร (document) มันย่อมมีความสำคัญลดลง Inverse Document Frequency มีวิธีคิดหลายวิธี เช่น smooth, max, probabilistic

### 2.2.2 RESTful API



#### ภาพที่ 2.5 RESTful API

https://i0.wp.com/saixiii.com/wp-content/uploads/2017/05/RESTful-API-logo-for-light-bg.png?fit=640%2C219&ssl=1

Representational state transfer หรือ REST คือ การสร้าง Webservice ชนิดหนึ่งที่ใช้สื่อสารกันบน Internet ใช้หลักการแบบ stateless คือไม่มี session ซึ่งต่างจาก webservice แบบอื่นเช่น WSDL และ SOAP การทำงานของ RESTful Webservice จะอาศัย URI/URL ของ request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูป XML, HTML, JSON โดย response ที่ตอบกลับจะเป็นการยืนยันผลของคำสั่งที่ส่งมา และสามารถพัฒนาด้วยภาษา programming ได้หลากหลาย คำสั่งก็จะมีตาม HTTP verbs ซึ่งก็คือ

* GET ทำกการดึงข้อมูลภายใน URI ที่กำหนด
* POST สำหรับสร้างข้อมูล
* PUT ใช้แก้ไขข้อมูล
* DELETE สำหรับลบข้อมูล

REST ถูกตั้งขึ้นโดย Roy Fielding ในปี 2000 ที่ University of California, Irvine ซึ่งได้ทำการพัฒนาขึ้นมาควบคู่กับ HTTP1.1 และ Uniform Resource Identifiers (URI)

### 2.2.3 SPA (Single Page Application)

Single Page Application คือการเขียนเว็บโดยใช้ html แค่หน้าเดียว หลังจากโหลดหน้าเว็บเสร็จแล้วจะไม่มีการ refresh browser เพื่อเปิดหน้าใหม่ ยกตัวอย่างเช่นหน้าแรกของ facebook ที่เราสามารถทำอะไรได้หลายๆ อย่างในหน้าเดียวเช่น โพสต์ status หรือดู notification ว่ามีใครมาเม้นต์หรือไลค์ข้อความใหนบ้าง หรือ follow/unfollow ซ่อนโพสต์ที่ไม่ชอบก็ได้ในหน้าเดียว แต่ facebook ก็ยังไม่ใช่ single-page สังเกตุได้จากเมื่อเราคลิกเข้าไปดู profile ตัวเอง มันมีการ refresh browser เพื่อเข้าไปดู

* ข้อดีของ SPA  
  ต้องการให้ผู้ใช้อยู่บนหน้าเว็บนานๆ ซึ่งเมื่ออยู่นานเข้าผู้ใช้ก็จะมีโอกาสทำอย่างอื่นมากขึ้น ซึ่งเขาจะได้แสดงโฆษณาได้นานขึ้น แล้วเขาก็ได้เงินจากส่วนนี้และยังทำให้ผู้ใช้งานรู้สึกใช้เว็บแล้วลื่นไหลไม่มีสะดุด เช่นไม่ต้องรอโหลดหน้าเว็บเป็นต้น
* ข้อเสียของ SPA
  + Security เนื่องจากการทำ SPA เป็นการทำงานบนหน้า html เพียงหน้าเดียว เราจะต้องคุม Security ให้ดี ซึ่งไม่เหมือนกับการเขียนเว็บปกติที่เราสามารถพึ่งพา server ได้เต็มที่ หนึ่งในเรื่องนี้ก็แก้ได้โดยการใช้พวก token เช่น https://jwt.io/ เป็นต้น
  + Bookmark เนื่องจากเรามีแค่หน้าจอเดียวแต่ทำงานได้หลากหลาย ทำให้การตั้ง bookmark จึงทำได้แค่เพียงหน้าเว็บแรกเท่านั้น ไม่สามารถ bookmark การทำงานในหน้าจอที่ต้องการได้
  + SEO เนื่องจากการทำงานเป็นการใช้ javascript render ทำให้ bot ของ search engine ไม่สามารถอ่านเนื้อหาข้างในได้
  + ไม่ support browser รุ่นเก่า

## 2.3 งานวิจัยหรือระบบงานที่ใกล้เคียง

* การวิเคราะห์ข้อมูลล็อกของระบบตรวจ สอบตัวตนสำหรับเครือข่ายไร้สาย
  + เทคโนโลยีเครือข่ายไร้สายเป็นช่องทางที่สำคัญในการเข้าใช้งานอินเตอร์เน็ตในหลายๆ องค์กรเนื่องจาก
  + เทคโนโลยีเครือข่ายไร้สายเป็นช่องทางที่สำคัญในการเข้าใช้งานอินเตอร์เน็ตในหลายๆ องค์กรเนื่องจาก
  + การออกแบบเครือข่ายไร้สายมีงบลงทุนด้านโครงสร้างพื้นฐานน้อยกว่าเครือข่ายแบบมีสาย ผู้ใช้เข้าถึงได้ง่ายกว่า
  + เพิ่มกลไกการตรวจสอบตัวตนเพื่อความมั่นคง
  + ระบบตรวจสอบตัวตนโดยส่วนใหญ่มีระบบบันทึกกิจกรรมการทำงานของระบบ
  + โดยข้อมูลทั้งหมดจัดเก็บไว้ในฐานข้อมูล RDBMS และ ข้อมูลล็อก

(วิริยะ รอดโพธิ์ทอง, ผู้ช่วยศาสตราจารย์ ดร. ปราลี มณีรัตน์, ฉันทวุฒิ พืชพล, ดร. ชติพจน์ ศรีเมือง)

* User Requirement Solicitation for an Information Retrieval System Applied to Dutch Grey Literature in the Archaeology Domain

we present the results of user requirement solicitation for a search system of grey literature in archaeology, specifically Dutch excavation reports. This search system uses Named Entity Recognition and Information Retrieval techniques to create an effective and effortless search experience. Specifically, we used Conditional Random Fields to identify entities, with an average accuracy of 56%. This is a baseline result, and we identified many possibilities for improvement. These entities were indexed in ElasticSearch and a user interface was developed on top of the index. This proof of concept was used in user requirement solicitation and evaluation with a group of end users. Feedback from this group indicated that there is a dire need for such a system, and that the first results are promising.(Alex Brandsen,Suzan Verberne,Karsten Lambers,Karsten Lambers,Milco Wansleeben Journal of Computer Applications in Archaeology)

* Elasticsearch - Advantages of DistributedSearch Data

Elastic-search is a platform for distributed search and analysis of data in real time.Its popularity is due to its ease of use, powerful features, and scalability. Today I amgoing to brief about the advantages of using elastic-search. (Michelle Leo, Cryptex technology)

* The Rise of Elastic Stack

This documentation compares Elastic's Elasticsearch/Logstash/Kibana  
(ELK) Stack, an open source search engine, analytics and Log Management Software (LMS) under Apache v2 license, and other Enterprise licensed software, mainly Splunk from high-level viewpoint. To present in details the real-world usability and best practice compliance of ELK Stack and how ELK was able to convert big companies to use their stack. Uncover the potential growth of development for ELK given the conversion of these company and possible contribution.(Davvid Paulo Sy ,School of Computing and Information,  
Technologies ,Asia Pacific College, Makati, Philippines)

* Building an IoT Data Hub with Elasticsearch,Logstash and Kibana

The goal of this publication is to describe the implementation of an Elastisearch, Logstash and Kibana (ELK) stack to process IoT data. Although, those tools were designed to be used mainly for handling large number of log data, they can be applied also to store, search and visualise other type of information-including IoT data. In order to show practical implementation of the idea, selected devices installed in the building of ABB Corporate Research in Krakow has been used. In prepared system, real data generated by various subsystems in this building has been integrated into one Elasticsearch based solution for further processing. Later on, this data will be used to develop data analytics to extract and visualize meaningful insights about building operations. In addition, selected data is sent to the Azure cloud to use its abilities in big data processing and machine learning.( Marcin Bajer,ABB Corporate Research Center, Starowi´slna 13a, 31-038 Krakow, Poland)

# บทที่ 3

**การดำเนินงานและผล**

เสนอผลการทดลองโดยละเอียด การเสนออาจเป็นแบบบรรยาย และ/หรือใช้ตาราง และ/หรือภาพประกอบ ควรมีการจัดหมวดหมู่ให้ตรงกับวัตถุประสงค์

## 3.1 ขอบเขตการดำเนินงาน

### 3.1.1 คุณลักษณะของระบบ

web service ให้บริการ HTTP ที่สามารถถูกใช้โดยโปรแกรมประยุกต์เว็บบนเว็บเบราว์เซอร์ โปรแกรมประยุกต์บนเดสก์ท็อป หรือแอพพลิเคชั่นโมบาย ที่มีข้อมูลบ้านพัก ห้องเช่าต่างๆ ให้ผู้ใช้ได้ทำการค้นหาข้อมูลต่างๆเหล่านี้โดยวิธีการค้นหาที่หลากหลาย และแสดงผลข้อมูลการเปรียบเทียบการค้นหาข้อมูลต่างๆ ให้ผู้ใช้งาน

### 3.1.2 ฟังก์ชันงานของระบบ

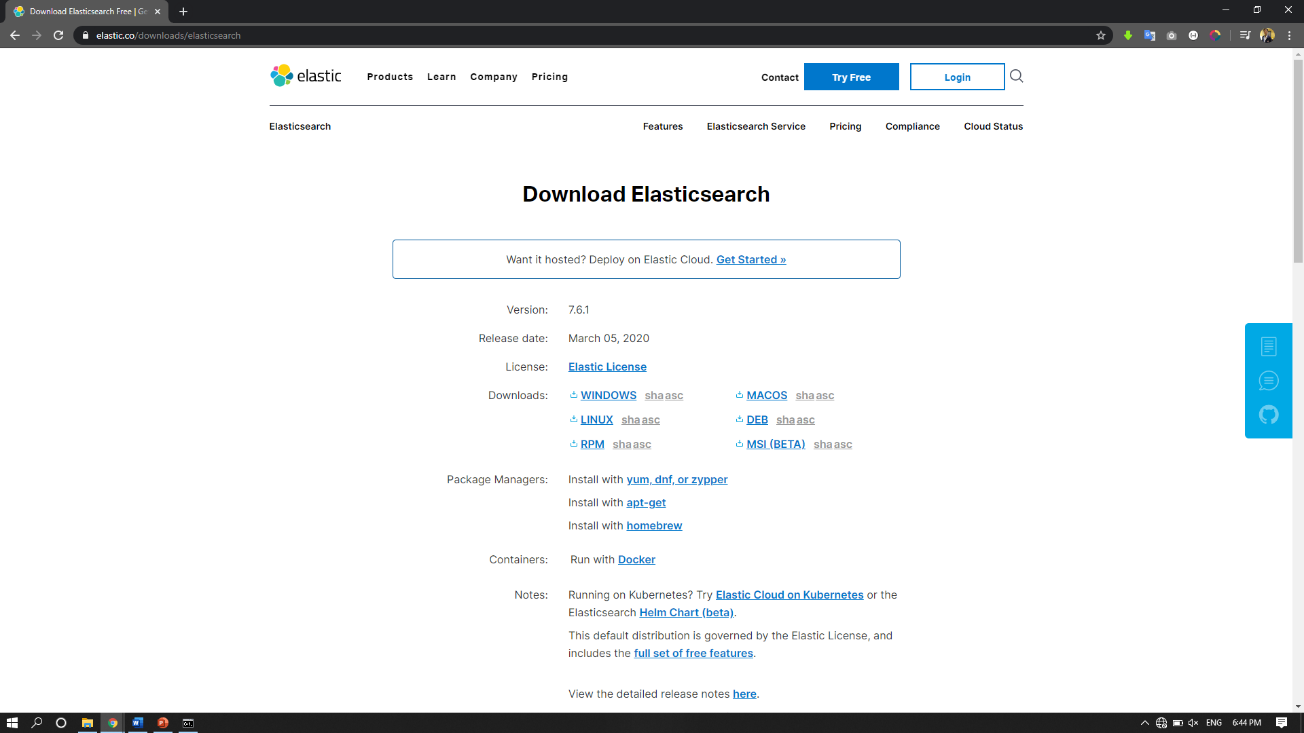
* ผู้ใช้งานสามารถค้นหาข้อมูลได้จากข้อมูลตัวอย่าง
* ผู้ใช้สามารถเลือกวิธีในการค้นหาข้อมูลได้
* ผู้ใช้สามารถดูรายละเอียดจากการค้นหาได้ เช่นความเร็วในการค้นหา เป็นต้น
* ผู้ใช้สามารถเพิ่มข้อข้อมูลได้

## 3.2 ขั้นตอนการพัฒนา

### 3.2.1 ขั้นตอนเตรียมการพัฒนา

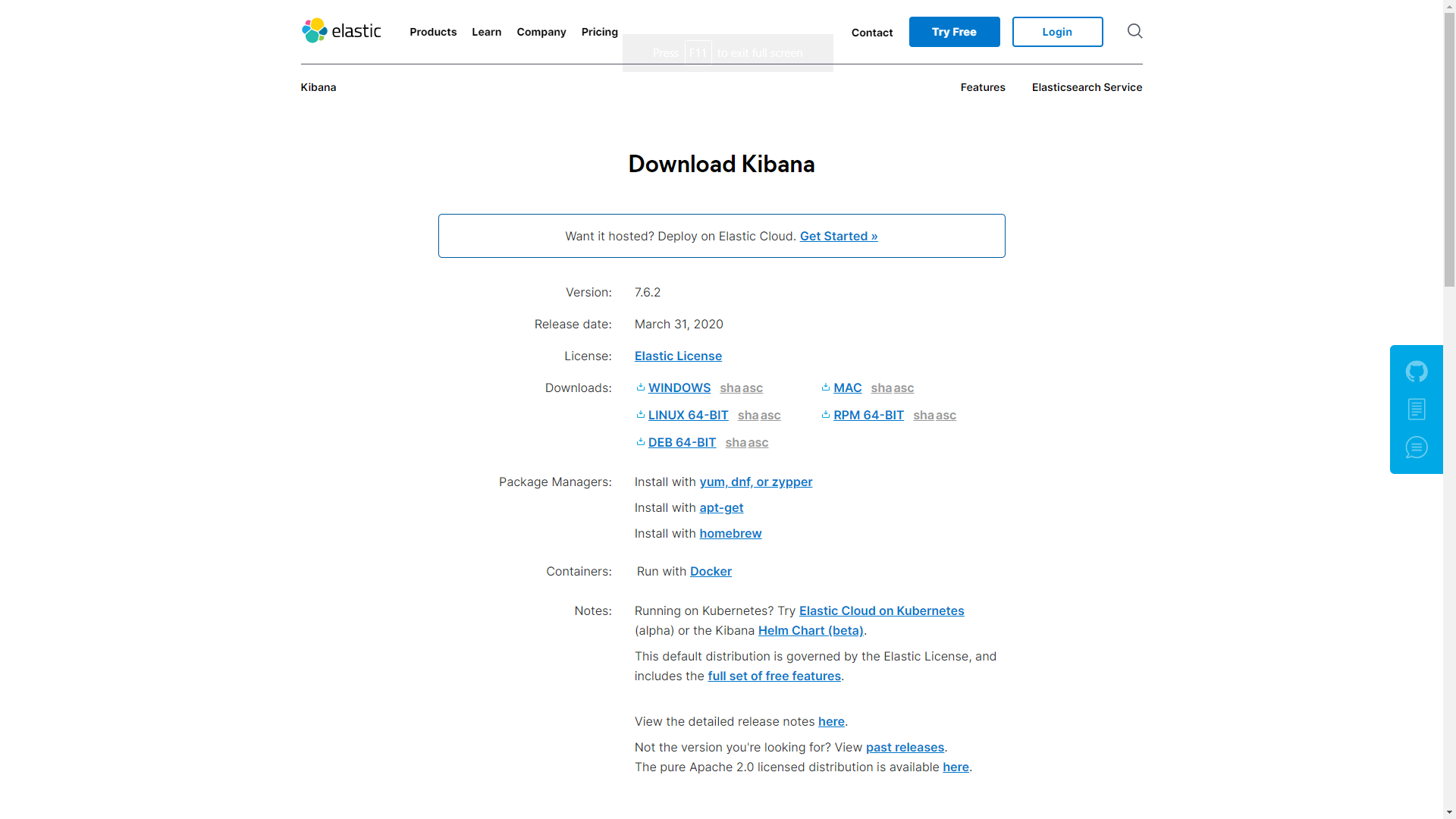
1. ติดตั้ง JDK 7 ขึ้นไป เช็ค version หากไม่ตรงตามกำหนดให้อับเดต

#### ภาพที่ 3.1 เช็ค version JDK

1. ****ติดตั้ง Elasticsearch Version : 7.6.1 ดาวโหลดจาก : https://www.elastic.co/  
   downloads/elasticsearch แล้ว Unzip file ไปใช้งาน

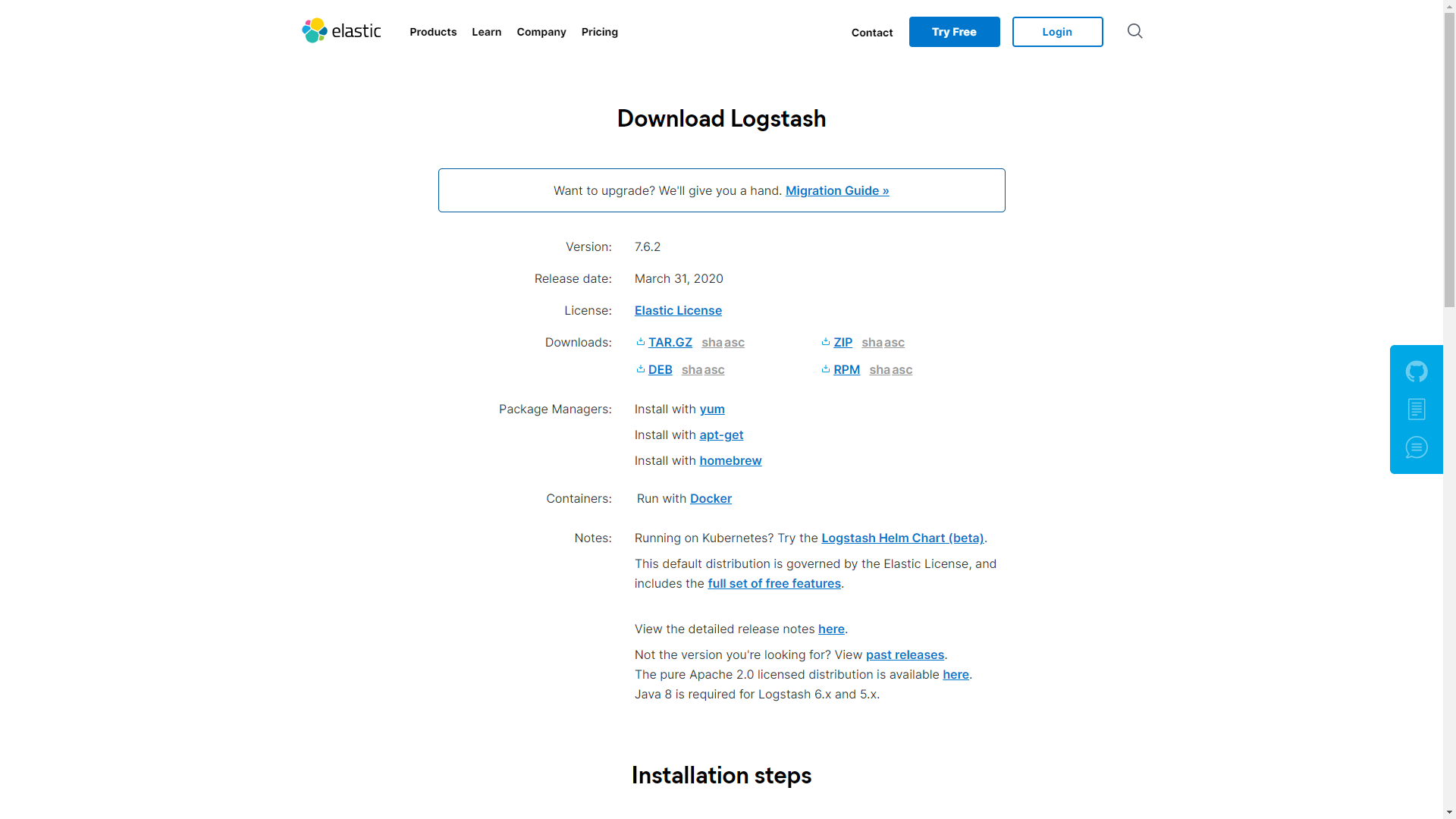
#### ภาพที่ 3.2 ดาวโหลด Elasticsearch

1. ติดตั้ง Kibana Version : 7.6.1 ดาวโลหดจาก : https://www.elastic.co  
   /downloads/kibana แล้ว Unzip file ไปใช้งาน



#### ภาพที่ 3.3 ดาวโหลด Kibana

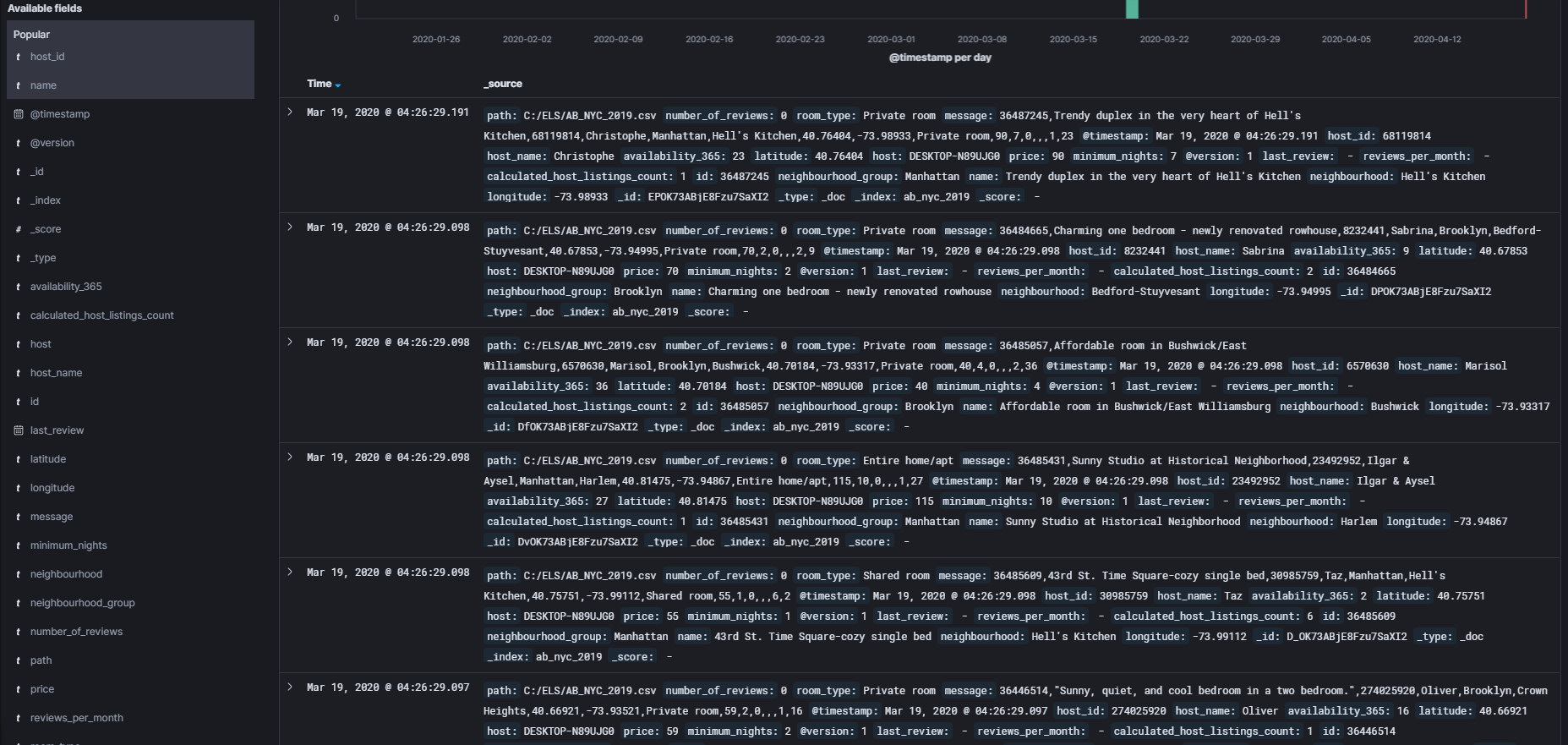
1. ติดตั้ง Logstash Version : 7.6.1 ดาวโลหดจาก : https://www.elastic.co  
   /downloads/kibana แล้ว Unzip file ไปใช้งาน



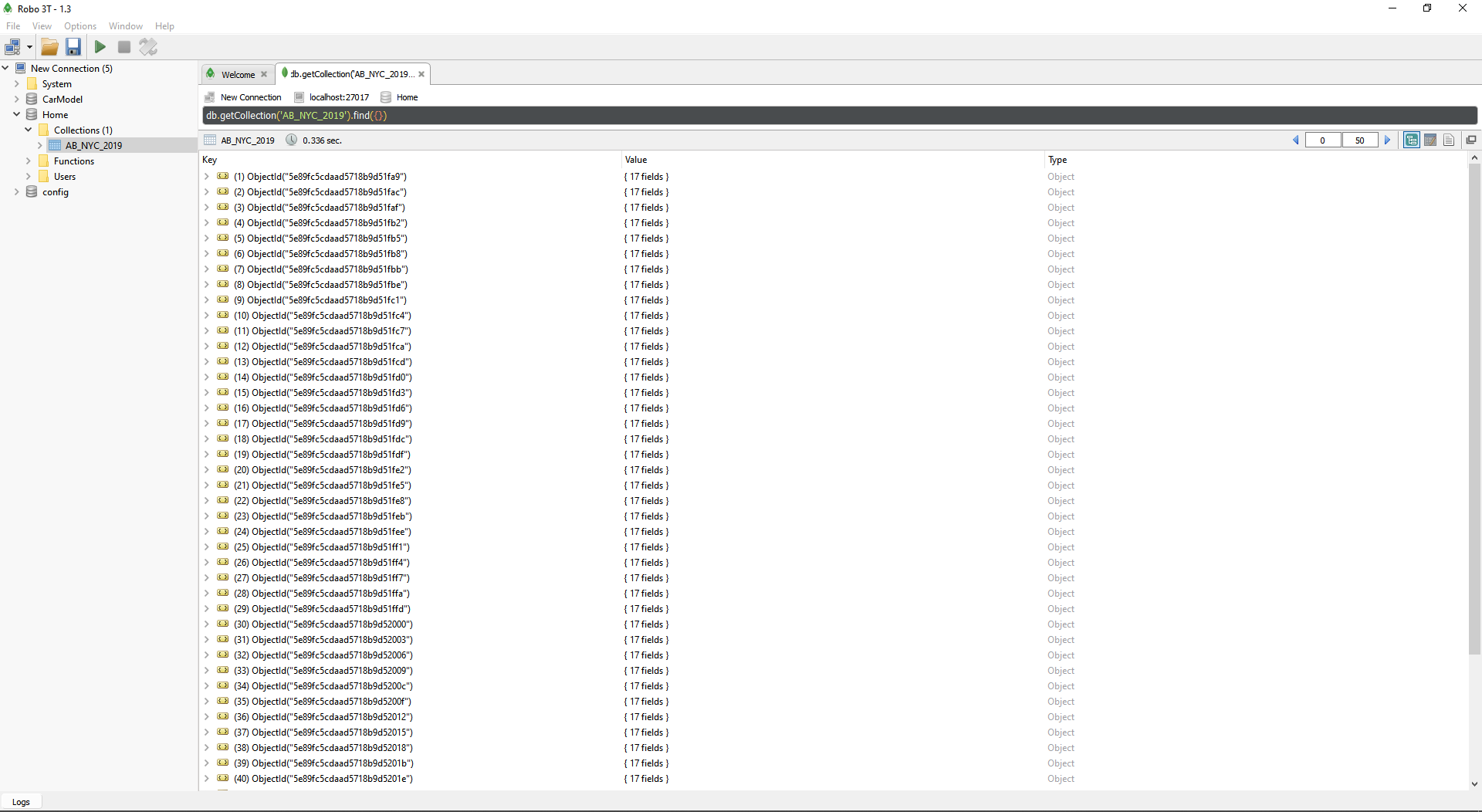
#### ภาพที่ 3.4 ดาวโหลด Logstash

### 3.2.2 ขั้นตอนการพัฒนา

- ฐานข้อมูลจะถูกสร้างขึ้นให้อัตโนมัติจากข้อมูลที่นำเข้า และยังมีฐานข้อมูลที่เอามาเปรียบเทียบซึ่งเป็น MySQL , MongoDB

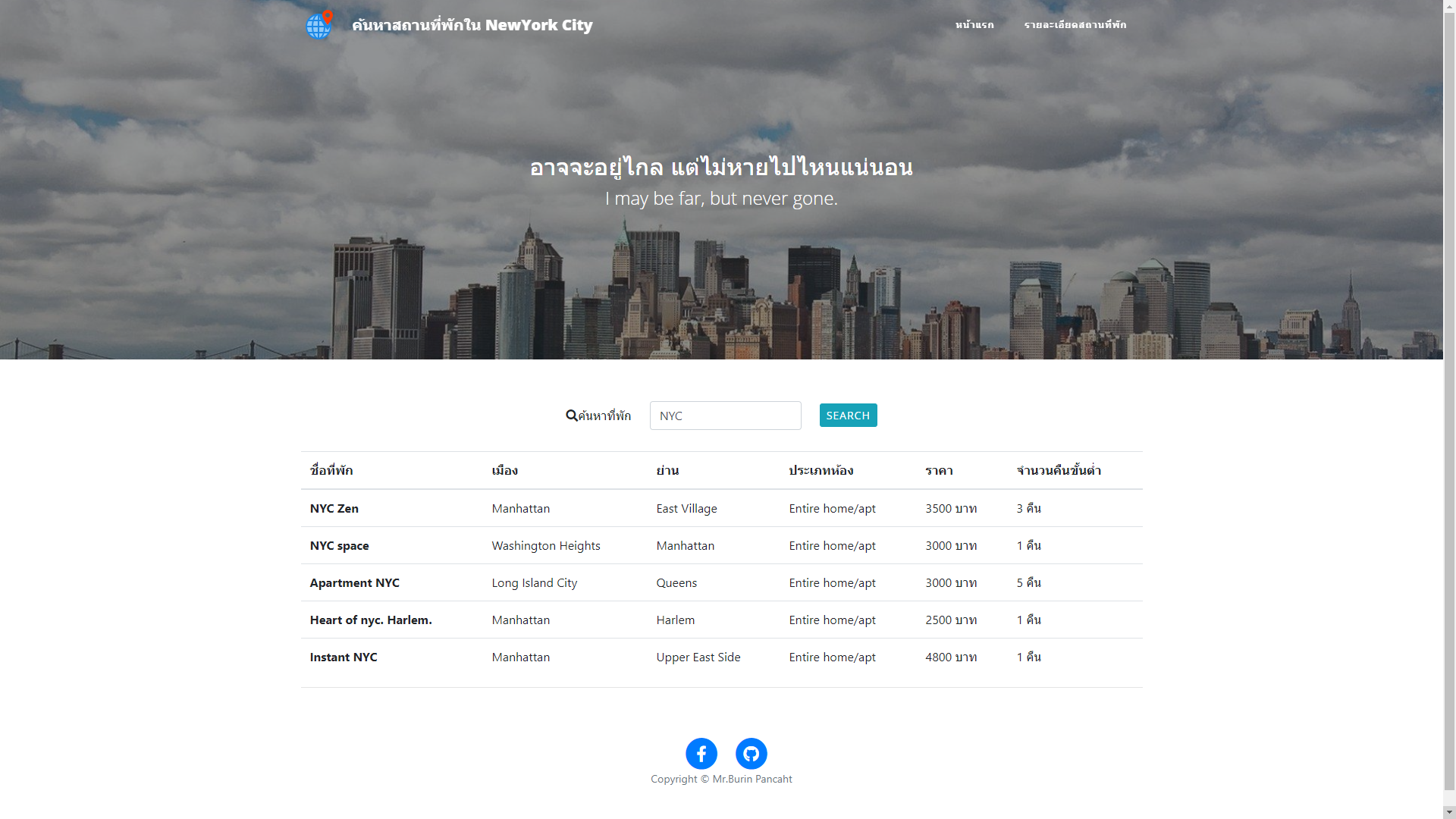


#### ภาพที่ 3.5 ภาพข้อมูล Elasticsearch



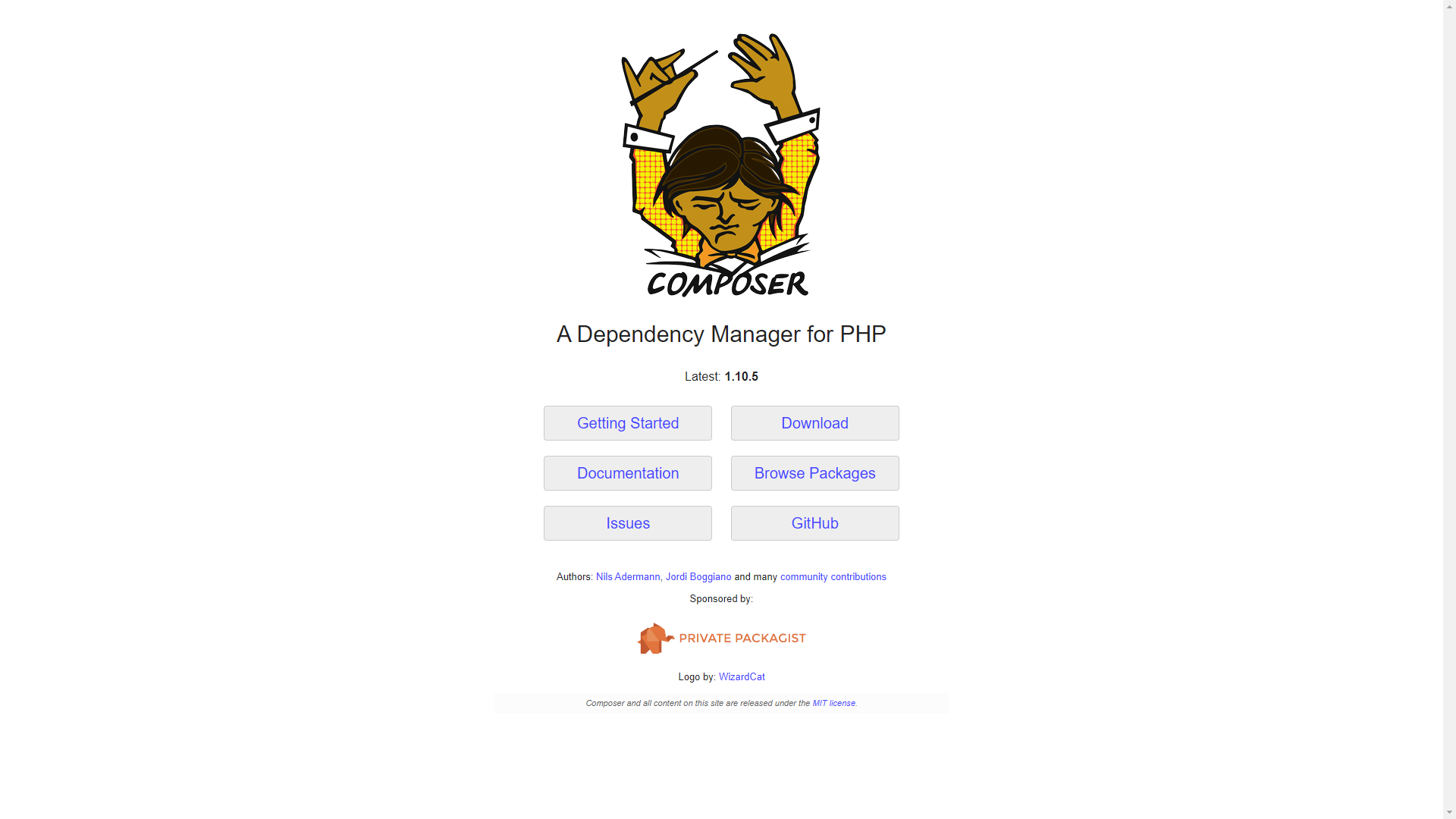
#### ภาพที่ 3.6 ภาพข้อมูล MongoDB

-หน้าจอตัวอย่างหน้าหลักในการค้นหา



#### ภาพที่ 3.7 ภาพตัวอย่างเว็บไซต์

-Code การติดตั้ง Composer ดาวโหลดจาก : https://getcomposer.org/

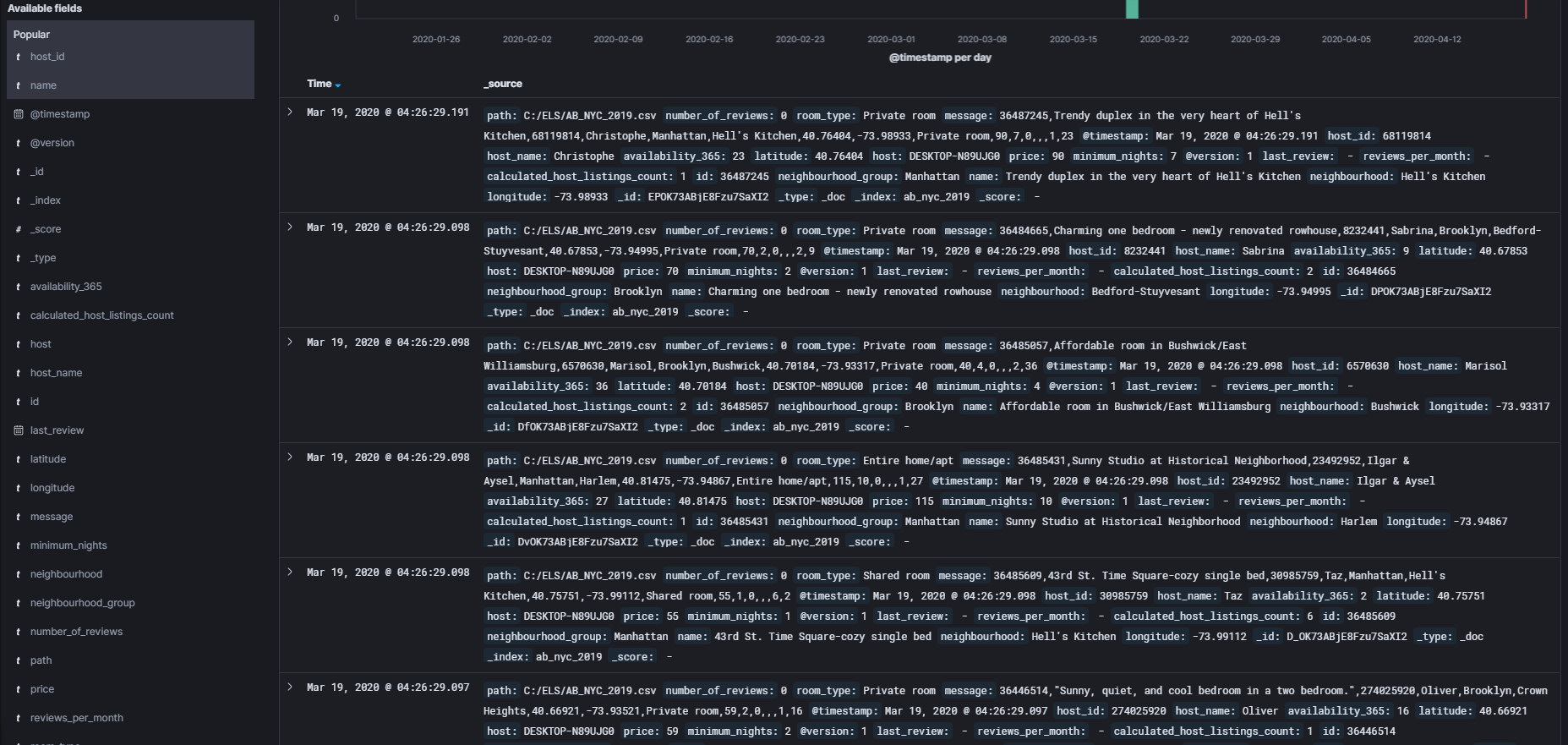


#### ภาพที่ 3.8 การดาวโหลดComposer



#### ภาพที่ 3.9 การติดตั้ง Composer

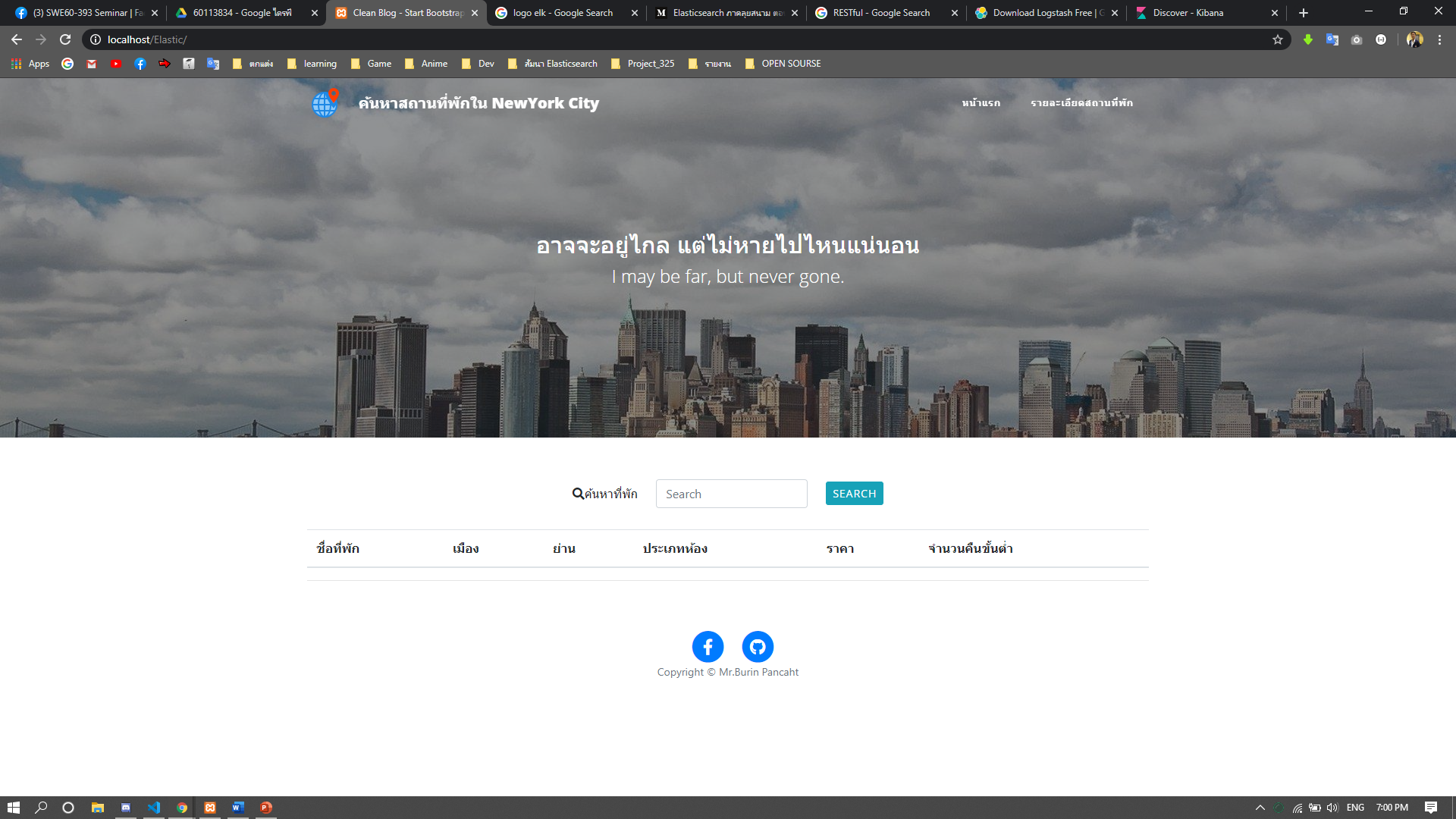
* + - 1. **การออกแบบฐานข้อมูล**



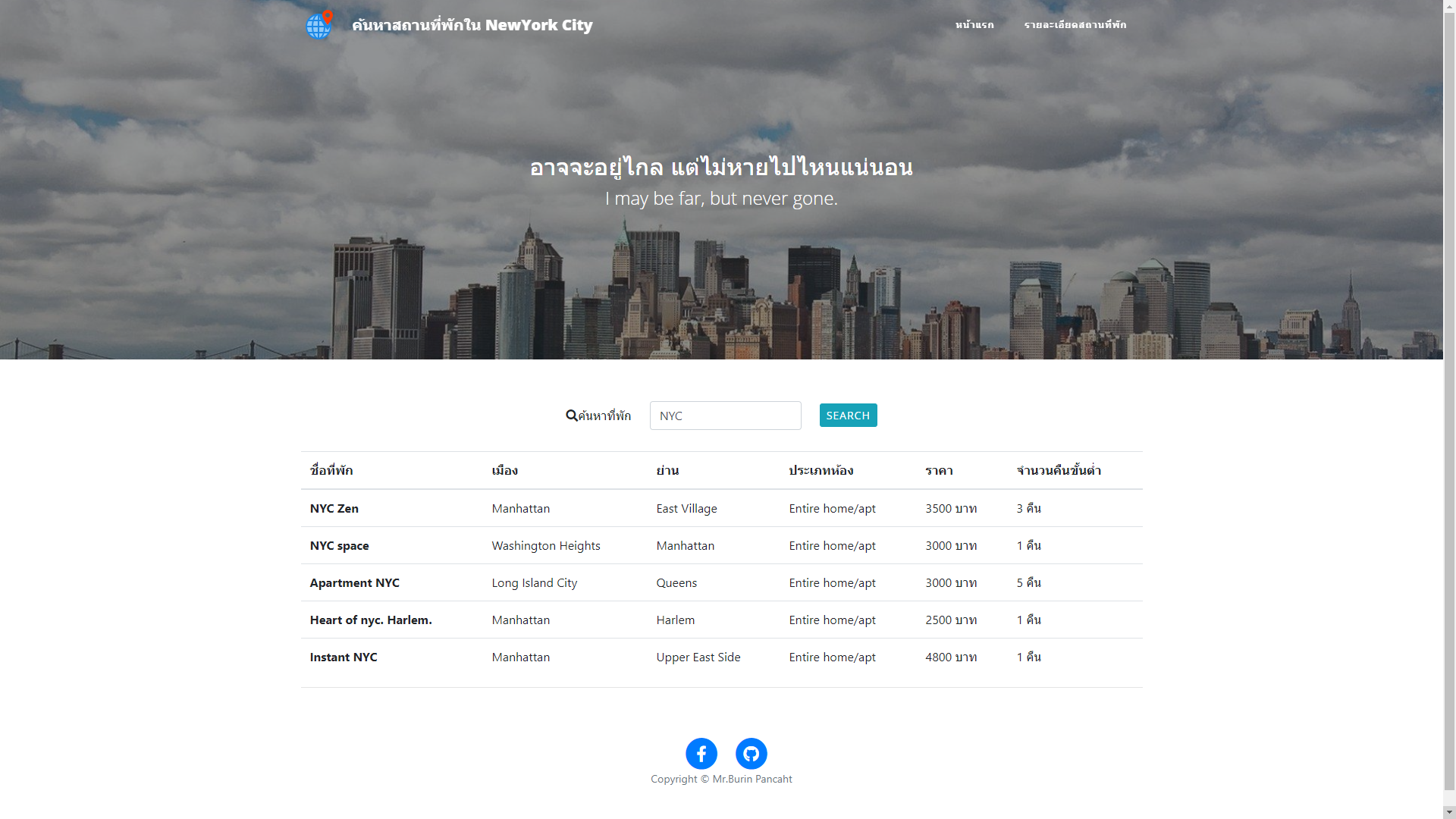
#### ภาพที่ 3.10 ข้อมูล

**ฐานข้อมูลจะถูกออกแบบให้โดยอันตโนมัติจากการนำ ข้อมูลเข้าไปใน Elasticsearch ผ่าน Logstash**

* + - 1. **แสดงการออกแบบ****หน้าจออินพุตเอาท์พุต**



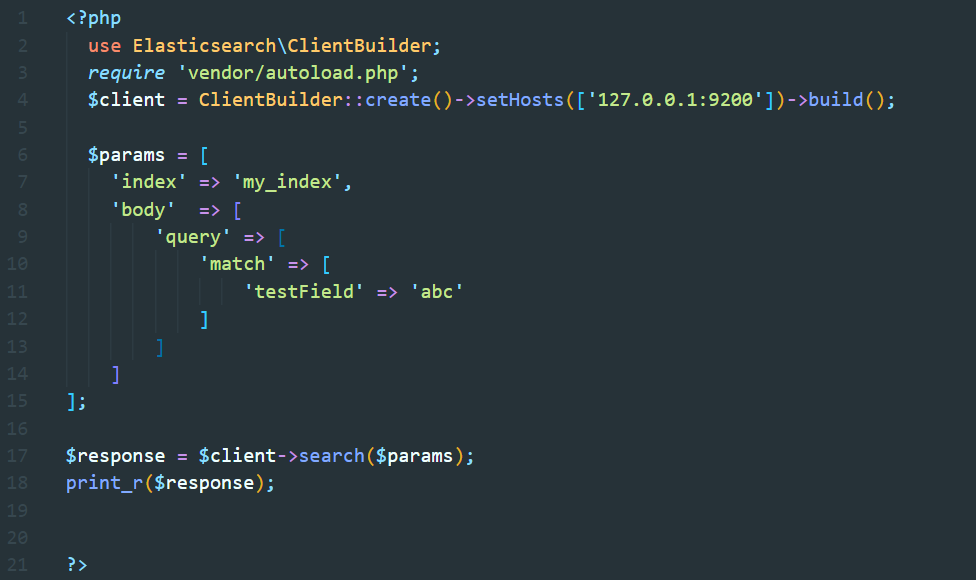
#### ภาพที่ 3.11 หน้าจออินพุต



#### ภาพที่ 3.12 หน้าจอเอาท์พุต

* + - 1. **ตัวอย่างการเขียนโปรแกรม**

-Code การconnect php กับ Elasticsearch



#### ภาพที่ 3.13 การconnect php กับ Elasticsearch

บรรทัดที่ 2 เรียกใช้งาน ClientBuilder

บรรทัดที่ 3 โหลด autoload เข้ามาใช้งาน

บรรทัดที่ 4 การ connect โดย set host เป็น Localhost port 9200

บรรทัดที่ 6-15 สร้างตัวแปรในการนำข้อมูลออกมา

บรรทัดที่ 17 เรียกใช้งานการค้นหาข้อมูลตามตัวแปรที่ส่งไป

บรรทัดที่ 18 แสดงผลข้อมูลที่ได้

-Code การนำข้อมูลเข้า Elasticsearch ด้วย Logstash จาก Logstashs.conf



#### ภาพที่ 3.14 Logstashs.conf

บรรทัดที่ 3 ที่อยู่ของไฟล์ข้อมูล

บรรทัดที่ 10 การกำหนดตัวที่ใช่แบ่งข้อมูล

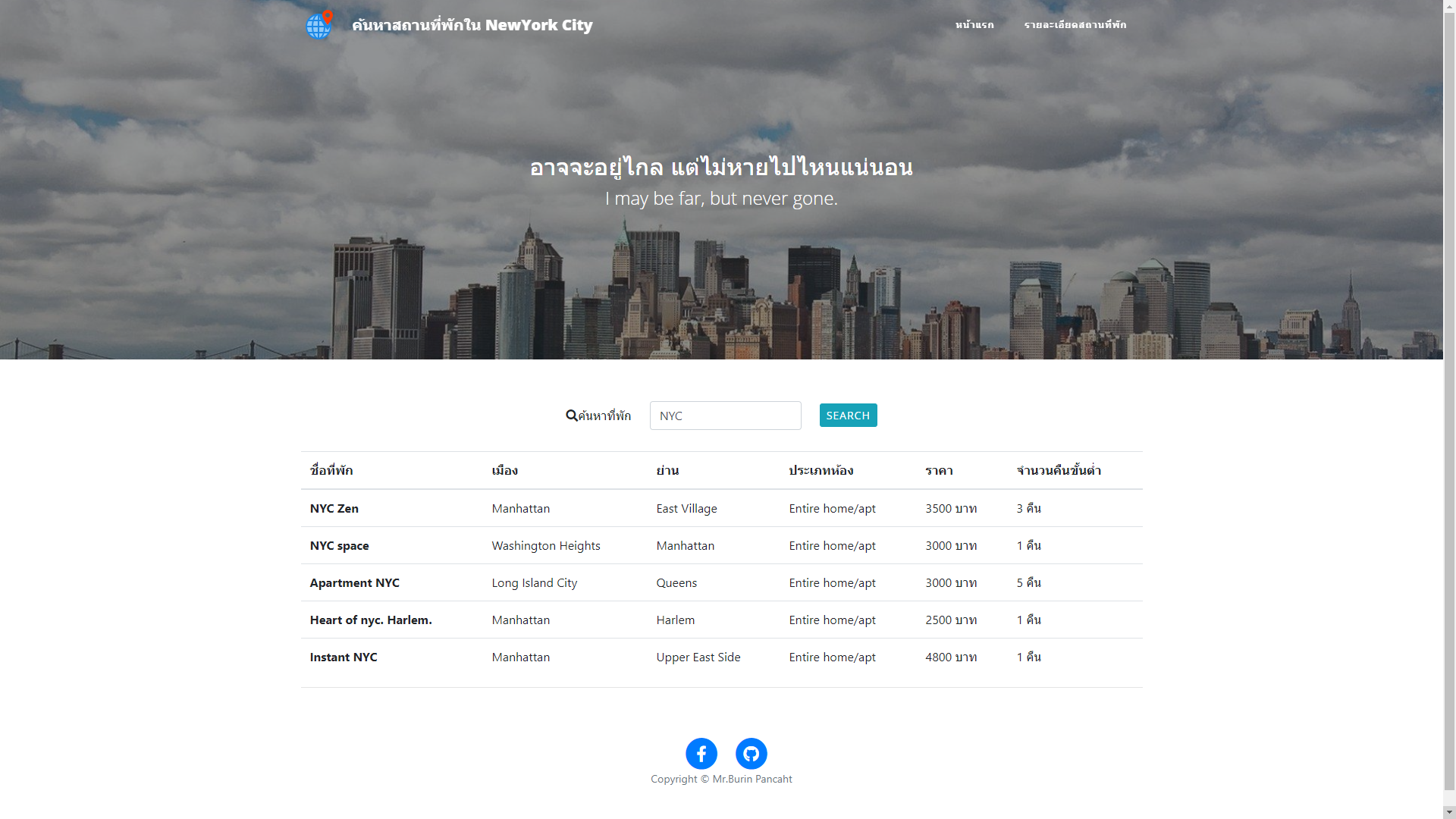
บรรทัดที่ 11-13 columns ที่ต้องการนำเข้าข้อมูล

บรรทัดที่ 18 นำข้อมูลไปไว้ที่ elasticsearch

บรรทัดที่ 19 สร้าง index ในการค้นหา

### 3.2.3ขั้นตอนการทดสอบ

**-**ค้นหาที่พักจากคำว่า nyc และแสดงผลข้อมูลออกมาจากคะแนนที่มากที่สุดที่ได้จากการค้นหาคำ



#### ภาพที่ 3.15 การค้นหาที่พักNYC

## 3.3 ผลการดำเนินงาน

ในหน้าแรกจะเป็นหน้าค้นหาที่ให้ผู้เข้าใช้งานสามารถค้นหาชื่อที่พักใน new York city ที่สามารถพิมพ์คำค้นหาได้ตามที่userต้องการค้นหาและจะแสดงข้อมูลที่ได้จากการค้นหาออกมาให้user ได้เห็นแต่จะมีการทำงานเบื้องหลังที่มีการค้นหาข้อมูลมีการทำคะแนนข้อมูลที่น่าจะตรงตามที่ลูกค้าต้องการ ด้วยการทำ TF-IDF

# บทที่ 4

**อภิปรายและสรุปผลการดำเนินงาน**

จากการศึกษาการทำงานของ Elasticsearch ทำให้ได้เรียนรู้เครื่องมือทั้งชุดที่เรียกว่า ELK ที่ประกอบไปด้วย Elasticsearch,kibana, Logstash ในการใช้งานเครื่องมือทั้ง 3 ตัวที่ค่อยทำงานร่วมกันและการเขียนวิธีการ นำข้อมูลออกมาจาก Elasticsearch ที่มีเทคนิควิธีการค้นหาที่สามารถทำได้อย่างหลากหลายตามวัตถุประสงค์ที่เราต้องการนำเสนอ

โดยสิ่งที่ได้จากการศึกษาภาคทฤษฎี และจากการทดลองพัฒนาระบบด้วย Elasticsearch ทำให้รู้ถึงหลักการทำงานของ Elasticsearch ในการคำนวณหาคะแนนที่ใช้ในการเลือกคำที่จะนำมาแสดงว่าคำใหนควรแสดงก่อนหน้าหลังจากคำที่ผู้ใช่ต้องการ

สิ่งที่สามารถพัฒนาต่อได้ในอนาคต คือการนำเอาไปปรับใช้ในการพัฒนาการค้นหาที่มีประสิทธิภาพได้มากกว่าเดิม และฟังก์ชั่นการทำงานๆอื่นๆในการวิเคราะห์และประมวลผลออกมาให้สามารถเข้าใจข้อมูได้ง่ายยิ่งขึ้น และสิ่งที่ควรปรับปรุงจากการสัมมนาครั้งนี้ คือการทำความเข้าใจที่ต้องใช้เวลาในการศึกษาให้มากขึ้นกว่าเดิมและความพยายามให้การทดลองทดสอบสิ่งที่ไม่เคยได้สัมผัสและลองมาก่อน

# เอกสารอ้างอิง

* Edureka.(2019). ELK Stack Training & Certification.02/03/2020. retrieved from https://www.edureka.co/
* Shay Banon.(2020).Elastic Training.02/03/2020. retrieved from https://www.elastic.co/
* Thai Elastic User Group(2015).การสนทนา .25/03/2020. retrieved from https://web.facebook.com/groups/thaielastic
* elasticsearch-php(2019). Documentation.12/04/2020. retrieved from https://github.com/elastic/elasticsearch-php
* Elastic(2020).Elasticsearch Reference.02/03/2020 retrieved from https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html
* Prapeepat Uewichitrapochana Thinknet(2019). Elasticsearch Lesson 1 : Simple Thai search engine.22/03/2020. retrieved from https://blog.thinknet.co.th/tech/elasticsearch-lesson-1%C2%A0-simple-thai-search-engine