

StarGo

เว็บค้นหาสถานที่ Pet-Friendly

โดย

66070121 พงศธร สารีรูป

66070248 ชนิสรา จันท์คำจร

66070286 ปณัสยา บุญประกอบ

66070293 พงศ์ณวิชญ์ งามสวนพลู

66070294 พงศภัค ตั้งคอนันต์

โครงการนี้เป็นส่วนหนึ่งของวิชา

SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

แขนงวิชาวิศวกรรมซอฟต์แวร์

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2568

COPYRIGHT 2025

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อโครงการ	StarGo เว็บไซต์ค้นหาสถานที่ Pet-Friendly
นักศึกษา	นายพงศธร สารีรูป, นางสาวชนิสรา จันทร์คำจร, นางสาวปณัสยา บุญประกอบ, นายพงศ์ณวิทย์ งามสวนพลู, นายพงศภัค ตั้งคอนันต์
แขนงวิชา	วิศวกรรมซอฟต์แวร์
ปีการศึกษา	1/2568

บทคัดย่อ

ในปัจจุบัน การเลี้ยงสัตว์ได้รับความนิยมเพิ่มมากขึ้น สัตว์เลี้ยงได้กลายเป็นส่วนหนึ่งของครอบครัวที่เจ้าของให้ความรักและดูแลเอาใจใส่เป็นอย่างดี หนึ่งในกิจกรรมที่เจ้าของสัตว์เลี้ยงมักทำคือการพาออกไปทำกิจกรรมนอกบ้าน เช่น ท่องเที่ยว หรือพักผ่อนในสถานที่ต่าง ๆ อย่างไรก็ตาม การค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยง (Pet-Friendly) ยังเป็นเรื่องที่ยุ่งยาก เนื่องจากข้อมูลกระจัดกระจายและไม่เป็นระบบ

จากปัญหาดังกล่าว คณะผู้จัดทำจึงได้พัฒนาเว็บไซต์ที่ช่วยให้สามารถค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยงได้อย่างง่ายดาย และยังสามารถติดตามตารางสัตว์เลี้ยงได้ คณะผู้จัดทำมุ่งหวังให้เว็บแอปพลิเคชันนี้ช่วยลดความยุ่งยากในการค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยง (Pet-Friendly) และส่งเสริมให้การพาสัตว์เลี้ยงออกไปนอกบ้านเป็นเรื่องที่สนุกมากยิ่งขึ้น

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผศ.ดร.ทัศนัย พลอยสุวรรณ ที่ได้จัดการเรียนการสอน และถ่ายทอดความรู้ ซึ่งคณะผู้จัดทำได้นำมาประยุกต์ใช้ในการทำโครงการจนสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

คณะผู้จัดทำ

บทคัดย่อ.....	3
กิตติกรรมประกาศ	4
สารบัญ	6
บทที่ 1	8
1.1 ที่มาและความสำคัญ	8
1.2 วัตถุประสงค์และเป้าหมาย	8
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	9
1.4 กลุ่มผู้ใช้ (Target Users).....	9
บทที่ 2	10
2.1 Git.....	10
2.2 Github	10
2.3 Google cloud.....	10
2.4 Docker	10
2.5 ทฤษฎีการเรียนรู้ทางสังคม (Social Learning Theory)	10
2.6 ทฤษฎีการเลียนแบบ (Mimetic Theory)	11
2.7 แบบจำลองพฤติกรรมผู้บริโภคเชิงสังคมวิทยา (Sociological Model of Consumer Behavior) ..	11
บทที่ 3	12
3.1 ขั้นตอนการค้นหาความต้องการ	12
3.1.1 การสังเกตการณ์พฤติกรรมของผู้ใช้	12
3.1.2 แอปพลิเคชันต้นแบบ	12
3.2 ความต้องการของระบบ	12
3.2.1 Functional Specification.....	12
3.2.2 Non-Functional Specification	13
3.3 การออกแบบระบบ.....	13
3.3.1 Use-case Diagram	13
3.3.2 Use-case Description	14

3.3.3 ER diagram	20
3.3.4 Data dictionary	21
3.4 หลักการทำงานของระบบ.....	24
3.4.1 CI/CD Architecture.....	24
3.4.2 Load Balancing Architecture	25
บทที่ 4	26
หน้า ลงทะเบียน	26
หน้า เข้าสู่ระบบ.....	26
หน้า ค้นหาสถานที่.....	27
หน้า เพิ่มสถานที่.....	27
หน้า ลบสถานที่.....	28
หน้า ค้นหาสัตว์เลี้ยง	28
หน้า เพิ่มรายชื่อสัตว์เลี้ยง	29
หน้า ลบสัตว์เลี้ยง.....	29
หน้า แก้ไขสัตว์เลี้ยงของตนเอง.....	30
หน้า จัดการโปรไฟล์	30
หน้า เปลี่ยนรหัสผ่าน	31
หน้า ระวังการใช้งานผู้ใช้	31
บทที่ 5	32
สรุปผลการดำเนินการ.....	32
ปัญหาและอุปสรรคที่พบ.....	32

สารบัญ

บทที่ 1

ที่มาและความสำคัญ

1.1 ที่มาและความสำคัญ

ปัจจุบันการเลี้ยงสัตว์ได้รับความนิยมอย่างแพร่หลาย และมีความนิยมเพิ่มมากขึ้นอย่างต่อเนื่อง สัตว์เลี้ยงกลายเป็นส่วนหนึ่งของครอบครัวที่เจ้าของจำเป็นต้องให้ความรักและดูแลเอาใจใส่เป็นอย่างดี ทั้งในด้านสุขภาพ ความเป็นอยู่ หนึ่งในนั้นรวมไปถึงการพาออกไปทำกิจกรรมนอกบ้าน โดยเฉพาะสัตว์บางประเภท เช่น สุนัข ที่จำเป็นต้องออกไปทำกิจกรรมกลางแจ้งเป็นประจำ อย่างไรก็ตาม สถานที่ที่เปิดให้สัตว์เลี้ยงสามารถเข้าได้ยังมีจำนวนไม่มากนัก เจ้าของจึงมักต้องค้นหาข้อมูลล่วงหน้าว่าสถานที่ใดบ้างที่อนุญาตให้นำสัตว์เลี้ยงเข้าได้ ซึ่งสร้างความยุ่งยากและเสียเวลาในการพาสัตว์เลี้ยงออกไปทำกิจกรรมนอกบ้าน

อีกทั้งในยุคปัจจุบัน สื่อออนไลน์ได้ส่งผลให้ “ดาราหรืออินฟลูเอนเซอร์สัตว์เลี้ยง” ได้รับความนิยมมากขึ้น ทำให้ผู้คนจำนวนไม่น้อยอยากติดตามหรือไปเยือนสถานที่ที่สัตว์เลี้ยงชื่อดังเคยไป เพื่อที่จะตามรอย หรือพาสัตว์เลี้ยงของตัวเองไปบ้าง

ด้วยเหตุนี้ ผู้จัดทำจึงได้พัฒนาเว็บแอปพลิเคชันที่ช่วยให้ผู้เลี้ยงสัตว์สามารถค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยง (Pet-Friendly) ได้อย่างสะดวกและรวดเร็ว ไม่ว่าจะเป็นสวนสาธารณะ คาเฟ่ ที่พัก หรือสถานที่ท่องเที่ยวต่าง ๆ ทำให้เว็บแอปพลิเคชันนี้ตอบโจทย์ผู้เลี้ยงสัตว์ในปัจจุบัน ช่วยลดความยุ่งยากในการค้นหาสถานที่ และส่งเสริมให้การพาสัตว์เลี้ยงออกไปนอกบ้านเป็นเรื่องที่สนุกมากยิ่งขึ้น

1.2 วัตถุประสงค์และเป้าหมาย

- เพื่อเป็นศูนย์รวมสถานที่สำหรับสัตว์เลี้ยง ช่วยให้ผู้เลี้ยงสัตว์ค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยงได้ง่ายยิ่งขึ้น
- เพื่อให้เจ้าของสัตว์เลี้ยงบันทึกและแบ่งปันประสบการณ์การท่องเที่ยวของสัตว์เลี้ยงได้ง่ายยิ่งขึ้น
- เพื่อให้ตามรอยดาราสัตว์เลี้ยงได้ง่ายยิ่งขึ้น

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- ผู้เลี้ยงสัตว์ค้นหาสถานที่ที่เป็นมิตรกับสัตว์เลี้ยงได้ง่ายยิ่งขึ้น
- เจ้าของสัตว์เลี้ยงสามารถบันทึกประสบการณ์การท่องเที่ยวของสัตว์เลี้ยงได้
- ตามรอยดาราสัตว์เลี้ยงได้ง่ายยิ่งขึ้น

1.4 กลุ่มผู้ใช้ (Target Users)

- ผู้ที่เลี้ยงสัตว์เลี้ยงที่ต้องการหาสถานที่สำหรับทำกิจกรรมนอกบ้านร่วมกับสัตว์เลี้ยง
- ผู้ที่เป็นแฟนคลับดาราสัตว์เลี้ยง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 Git

Git เป็นระบบควบคุมเวอร์ชัน (Version Control System) ที่ช่วยในการจัดการและติดตามการเปลี่ยนแปลงของซอร์สโค้ดในระหว่างการพัฒนาโปรแกรม และช่วยให้ทีมสามารถทำงานร่วมกันบนโค้ดชุดเดียวกันได้อย่างเป็นระบบ โดยไม่เกิดการทับซ้อนของข้อมูล

2.2 Github

GitHub เป็นแพลตฟอร์มออนไลน์ที่ใช้สำหรับจัดเก็บและจัดการซอร์สโค้ดที่อยู่ภายใต้ระบบ Git โดย GitHub ช่วยให้ทีมสามารถทำงานร่วมกันบนโปรเจกต์เดียวกันได้อย่างมีประสิทธิภาพ

2.3 Google cloud

Google Cloud เป็นบริการคลาวด์คอมพิวติงจากบริษัท Google ที่ให้บริการด้านการประมวลผล จัดเก็บข้อมูล และโฮสต์เว็บไซต์บนระบบคลาวด์ ใช้ในการรันเซิร์ฟเวอร์ จัดการฐานข้อมูล หรือเก็บไฟล์ต่าง ๆ ทำให้ระบบสามารถทำงานได้อย่างต่อเนื่องและเข้าถึงได้จากทุกที่

2.4 Docker

Docker เป็นเทคโนโลยีที่ใช้สำหรับสร้าง "Container" ซึ่งเป็นสภาพแวดล้อมจำลองที่รวมทุกองค์ประกอบที่จำเป็นต่อการรันแอปพลิเคชันให้อยู่ภายในพื้นที่เดียวกัน การใช้ Docker ช่วยให้สามารถพัฒนาและทดสอบระบบด้วยสภาพแวดล้อมที่เหมือนกันทุกเครื่อง ลดปัญหาความไม่เข้ากันของระบบ และทำให้การนำเว็บไซต์ขึ้นใช้งานจริง (Deployment) เป็นไปอย่างรวดเร็วและมีประสิทธิภาพ

2.5 ทฤษฎีการเรียนรู้ทางสังคม (Social Learning Theory)

ทฤษฎีของอัลเบิร์ต แบนดูรา (Albert Bandura) ระบุว่ามนุษย์เรียนรู้พฤติกรรมต่างๆ จากการสังเกตและเลียนแบบผู้อื่นในสังคม โดยเฉพาะบุคคลที่เป็น "ตัวแบบ" (Model) ผู้คนจึงเลียนแบบสไตล์การแต่งตัว การใช้ผลิตภัณฑ์ หรือแม้กระทั่งไลฟ์สไตล์ของเซเลบ เพราะมองว่าเซเลบเป็น "ตัวแบบ" ที่น่าเอาเป็นแบบอย่าง และมองว่าประสบความสำเร็จ

2.6 ทฤษฎีการเลียนแบบ (Mimetic Theory)

เรอเน่ จิราร์ด (René Girard) นักจิตวิทยาชาวฝรั่งเศสเป็นผู้ริเริ่มทฤษฎีนี้ โดยอธิบายว่าความปรารถนาของมนุษย์ส่วนใหญ่มักไม่ใช่ความต้องการที่เกิดขึ้นเอง แต่เป็นการเลียนแบบความต้องการของผู้อื่นที่เราต้องการสิ่งใดสิ่งหนึ่งเป็นเพราะเห็นคนอื่นต้องการสิ่งนั้น

ปรากฏการณ์ที่สินค้าชิ้นหนึ่งกลายเป็นที่ต้องการอย่างสูงในทันที หลังจากมีเซเลบหรือคนดังนำมาใช้ เพราะผู้คนไม่ได้ต้องการสินค้านั้นโดยตรง แต่ต้องการ "เลียนแบบ" ความปรารถนาของคนดังที่ตนชื่นชอบ

2.7 แบบจำลองพฤติกรรมผู้บริโภคเชิงสังคมวิทยา (Sociological Model of Consumer Behavior)

แบบจำลองนี้เน้นว่าพฤติกรรมการตัดสินใจซื้อของผู้บริโภคได้รับอิทธิพลอย่างมากจากปัจจัยทางสังคม และกลุ่มที่ตนเองเป็นสมาชิกอยู่ ซึ่งรวมถึง ครอบครัว, เพื่อน, วัฒนธรรม, และชนชั้นทางสังคม การตามเทรนด์จึงไม่ใช่แค่เรื่องของปัจเจกบุคคล แต่เป็นผลมาจากแรงกดดันและความคาดหวังจากกลุ่มสังคมต่าง ๆ ที่รายล้อมตัวเอง

บทที่ 3

ขั้นตอนการดำเนินงานของโครงการ

3.1 ขั้นตอนการค้นหาคำความต้องการ

3.1.1 การสังเกตการณ์พฤติกรรมของผู้ใช้

ผู้จัดทำได้เห็นถึงปัญหาในการรวบรวมข้อมูลร้านอาหารและสถานที่ต่าง ๆ ซึ่งพบเจอจากการสังเกตการณ์พฤติกรรมของบุคคลในโลกออนไลน์ตามแพลตฟอร์มต่าง ๆ โดยมีรายละเอียด ดังนี้

- **ขาดศูนย์กลางข้อมูล** เนื่องจากไม่มีแพลตฟอร์มใดที่ถูกสร้างมาเพื่อรวบรวมข้อมูลประเภทนี้โดยเฉพาะ
- **การสอบถามข้อมูลซ้ำซ้อน** ทั้งในกระทู้ กลุ่ม LINE OpenChat และ ช่องทางอื่น ๆ
- **ภาระในการแบ่งปันข้อมูล** ของแฟนคลับที่ต้องคอยตอบคำถามซ้ำซ้อน

3.1.2 แอปพลิเคชันต้นแบบ

โครงการนี้มีแอปพลิเคชันต้นแบบ ชื่อ BINGO-K-pop Idols Restaurant ซึ่งเป็นแอปพลิเคชันที่รวบรวมร้านอาหารยอดนิยมที่เหล่าไอดอล K-pop ได้ไปเยือน โดยมุ่งเน้นไปที่บุคคลสาธารณะในประเทศเกาหลี (https://play.google.com/store/apps/details?id=com.bingo.bingo.bingo_application)

3.2 ความต้องการของระบบ

3.2.1 Functional Specification

ระบบจะมี 2 actors ด้วยกัน ได้แก่

- **User** คือ บุคคลทั่วไป
 - บุคคลทั่วไปต้องสามารถลงทะเบียน เข้าสู่ระบบ และเปลี่ยนรหัสผ่านบัญชีได้
 - ผู้ใช้ที่ล็อกอินแล้วต้องสามารถเพิ่มรายชื่อสัตว์เลี้ยงที่มีชื่อเสียง สัตว์เลี้ยงของตนเอง และเพิ่มสถานที่ได้
 - ผู้ใช้ที่ล็อกอินแล้วต้องสามารถค้นหาบุคคลสาธารณะและสถานที่ได้
- **Admin** คือ ผู้ดูแลระบบ
 - ผู้ดูแลระบบต้องสามารถเพิ่มรายชื่อสัตว์เลี้ยงที่มีชื่อเสียง สัตว์เลี้ยงของตนเอง และเพิ่มสถานที่ได้
 - ผู้ดูแลระบบต้องสามารถลบบุคคลสาธารณะ ลบสถานที่ และระงับการใช้งานผู้ใช้ได้

3.2.2 Non-Functional Specification

- ด้านความปลอดภัย (Security) ข้อมูลรหัสผ่านของผู้ใช้จะต้องถูกเข้ารหัสก่อนจัดเก็บลงฐานข้อมูล

3.3 การออกแบบระบบ

3.3.1 Use-case Diagram



3.3.2 Use-case Description

Use-case: ลงทะเบียน

Use-Case Name	ลงทะเบียน	
Actors	User	
Brief Description	ผู้ใช้สร้างบัญชีเพื่อใช้งานแพลตฟอร์ม	
Flows of events	User	System
	1. เข้าหน้าลงทะเบียน 2. กรอกข้อมูลตามที่ระบบต้องการ 3. กดปุ่มเพื่อส่งข้อมูล	4. ระบบสร้างบัญชีและพาเข้าสู่ระบบ
Alternative flow	-	
Pre-Condition	ผู้ใช้ต้องสร้างบัญชีก่อน	
Post-Condition	-	

Use-case: เข้าสู่ระบบ

Use-Case Name	เข้าสู่ระบบ	
Actors	User, Admin	
Brief Description	ผู้ใช้เข้าสู่ระบบเพื่อใช้งานแพลตฟอร์ม	
Flows of events	User	System
	1. เข้าหน้า Login 2. ผู้กรอก Username และ Password ให้ถูกต้อง	3. ระบบพาเข้าสู่ระบบ
Alternative flow	-	
Pre-Condition	ผู้ใช้ต้องสร้างบัญชีก่อน	
Post-Condition	-	

Use-case: ค้นหาสถานที่

Use-Case Name	ค้นหาสถานที่	
Actors	User, Admin	
Brief Description	ผู้ใช้ค้นหาสถานที่	
Flows of events	User	System
	1. พิมพ์ชื่อสถานที่ 2. กดเลือกสถานที่	
		3. ระบบนำเข้าสู่หน้าสถานที่
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: เพิ่มสถานที่

Use-Case Name	เพิ่มสถานที่	
Actors	User, Admin	
Brief Description	เมื่อผู้ใช้เพิ่มสถานที่แล้ว สถานที่จะแสดงให้ผู้ใช้ทุกคนได้เห็น	
Flows of events	User	System
	1. กดเพิ่มสถานที่ 2. กรอกชื่อ พิกัด ที่อยู่ และเพิ่มรูปภาพ ของสถานที่ 3. กด Confirm	
		4. ระบบบันทึกข้อมูลของสถานที่ ลงฐานข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: ลบสถานที่

Use-Case Name	ลบสถานที่	
Actors	Admin	
Brief Description	Admin สามารถเลือกลบสถานที่ได้ ที่ Django admin	
Flows of events	User	System
	1. เข้าไปที่หน้าการจัดการสถานที่ 2. เลือกลบสถานที่	
		3. ระบบลบสถานที่ออกจากฐานข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้งานต้องเข้าสู่ระบบด้วยบัญชี admin ก่อน	
Post-Condition		

Use-case: ค้นหาสัตว์เลี้ยง

Use-Case Name	ค้นหาสัตว์เลี้ยง	
Actors	User, Admin	
Brief Description	ผู้ใช้งานสามารถค้นหาสัตว์เลี้ยงได้	
Flows of events	User	System
	1. พิมพ์ชื่อสัตว์เลี้ยงที่ต้องการค้นหา 2. เลือกสัตว์เลี้ยง	
		3. ระบบนำผู้ใช้งานไปที่หน้าโปรไฟล์สัตว์เลี้ยง
Alternative flow		
Pre-Condition	ผู้ใช้งานต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: เพิ่มรายชื่อสัตว์เลี้ยง

Use-Case Name	เพิ่มรายชื่อสัตว์เลี้ยง	
Actors	User, Admin	
Brief Description	ผู้ใช้เพิ่มรายชื่อสัตว์เลี้ยงเข้าระบบได้ โดยอาจเป็นสัตว์เลี้ยงของตนเอง หรือสัตว์เลี้ยงที่มีชื่อเสียง	
Flows of events	User	System
	1. กดปุ่มเพิ่มรายชื่อสัตว์เลี้ยง 2. กรอกข้อมูลตามที่ระบบต้องการ 3. กดส่งเข้าระบบ	
		4. ระบบบันทึกข้อมูลเข้าฐานข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: ลบสัตว์เลี้ยง

Use-Case Name	ลบสัตว์เลี้ยง	
Actors	Admin	
Brief Description	ผู้ใช้ที่เป็น Admin ลบสัตว์เลี้ยงออกจากระบบ	
Flows of events	User	System
	1. เข้าไปที่หน้าการจัดการสัตว์เลี้ยง 2. เลือกลบสัตว์เลี้ยง	
		3. ระบบบันทึกข้อมูลเข้าฐานข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบด้วยบัญชี Admin	
Post-Condition		

Use-case: แก้ไขส่วนตัวเลี้ยงของตนเอง

Use-Case Name	แก้ไขส่วนตัวเลี้ยงของตนเอง	
Actors	User, Admin	
Brief Description	ผู้ใช้ที่เพิ่มส่วนตัวเลี้ยงของตนเองในระบบสามารถแก้ไขข้อมูลได้	
Flows of events	User	System
	1. ผู้ใช้ไปที่หน้า Profile 2. เลือกส่วนตัวเลี้ยงเลี้ยงของตนเองที่ต้องการแก้ไขข้อมูล 3. แก้ไขข้อมูล	
		4. ระบบอัปเดตข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: จัดการโปรไฟล์

Use-Case Name	จัดการโปรไฟล์	
Actors	User	
Brief Description	ผู้ใช้เปลี่ยนหรือลบรูปโปรไฟล์ ชื่อ นามสกุล และ ชื่อผู้ใช้ได้	
Flows of events	User	System
	1. ผู้ใช้ไปที่หน้า Edit profile 2. เลือกแก้ไขชื่อนามสกุล ชื่อผู้ใช้และรูปโปรไฟล์	
		3. ระบบอัปเดตข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้ต้องเข้าสู่ระบบก่อน	
Post-Condition		

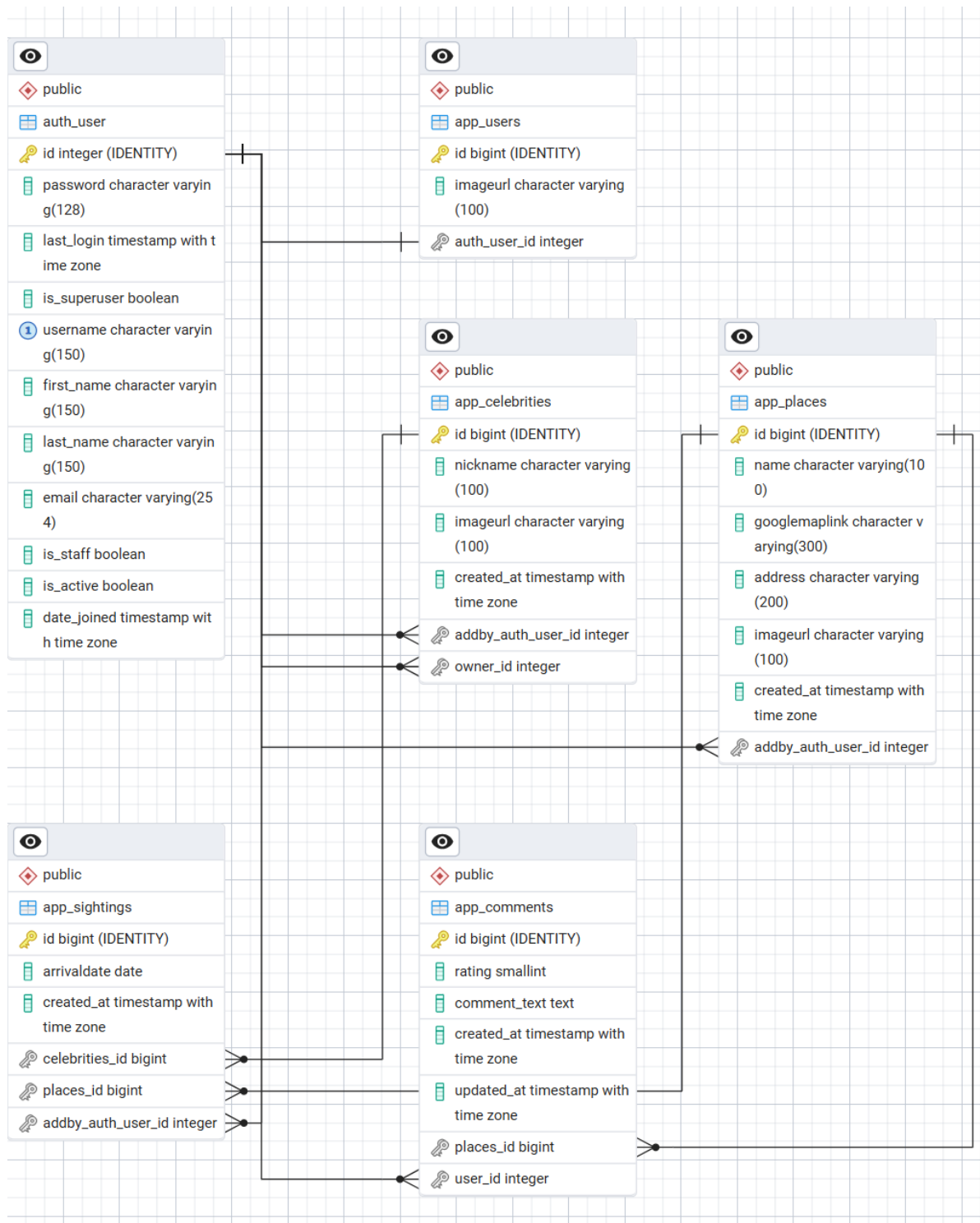
Use-case: เปลี่ยนรหัสผ่าน

Use-Case Name	เปลี่ยนรหัสผ่าน	
Actors	User, Admin	
Brief Description	ผู้ใช้เปลี่ยนรหัสผ่าน	
Flows of events	User	System
	1. ผู้ใช้ไปที่หน้า Change password	
	2. กรอกรหัสเก่า รหัสใหม่ และยืนยันรหัส	
	3. กดส่งเข้าระบบ	
		4. ระบบอัปเดตข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้งานต้องเข้าสู่ระบบก่อน	
Post-Condition		

Use-case: ระงับการใช้งานผู้ใช้

Use-Case Name	ระงับการใช้งานผู้ใช้	
Actors	Admin	
Brief Description	ผู้ใช้เลือกระงับการใช้งานผู้ใช้ที่มีพฤติกรรมไม่ดี หรือบัญชีที่มีความผิดปกติ	
Flows of events	User	System
	1. เข้าไปที่หน้าการจัดการบัญชี user	
	2. เลือก is_active เป็น false	
	3. กดบันทึกการเปลี่ยนแปลง	
		4. ระบบอัปเดตข้อมูล
Alternative flow		
Pre-Condition	ผู้ใช้งานต้องเข้าสู่ระบบด้วยบัญชี Admin	
Post-Condition		

3.3.3 ER diagram



.3.3.4 Data dictionary

แพลตฟอร์มนี้ถูกสร้างขึ้นโดยใช้ Django ซึ่งเป็น Framework ที่มีฟังก์ชันสำหรับสร้างตารางเพื่อจัดการสิทธิ์ผู้ใช้งาน (Authentication) มาให้เป็นมาตรฐาน ในเอกสารนี้จึงจะกล่าวถึงเฉพาะตารางที่ผู้พัฒนาได้สร้างขึ้นใหม่ และตารางส่วนของ Django ที่ถูกนำมาใช้งานในระบบเท่านั้น โดย ตารางที่มีคำว่า “auth_” อยู่หน้าชื่อตาราง คือตารางที่ Django สร้างขึ้น

Table app_celebrities

Column Name	Data type	Description
id	BIGINT	Primary Key ของสัตว์เลื้อยในระบบ
nickname	VARCHAR(100)	ชื่อเล่นสัตว์เลื้อย
imageurl	VARCHAR(100)	URL ของรูปภาพสัตว์เลื้อย
created_at	TIMESTAMP	วันที่และเวลาที่เพิ่มข้อมูล
addby_auth_user_id	INTEGER	รหัสผู้ใช้ที่เพิ่มข้อมูลสัตว์เลื้อยนี้
owner_id	INTEGER	รหัสผู้ใช้ที่เป็นเจ้าของสัตว์เลื้อย

Table app_places

Column Name	Data type	Description
id	BIGINT	Primary Key ของสถานที่
name	VARCHAR(100)	ชื่อสถานที่
googlemaplink	VARCHAR(300)	ลิงก์ Google Maps ของสถานที่
address	VARCHAR(200)	ที่อยู่ของสถานที่
imageurl	VARCHAR(100)	URL ของรูปภาพสถานที่
created_at	TIMESTAMP	วันที่และเวลาที่เพิ่มข้อมูล
addby_auth_user_id	INTEGER	รหัสผู้ใช้ที่เพิ่มข้อมูลสถานที่นี้

Table app_users

Column Name	Data type	Description
id	BIGINT	Primary Key ของตารางโปรไฟล์
imageurl	VARCHAR(100)	URL ของรูปภาพโปรไฟล์
auth_user_id	INTEGER	Foreign Key อ้างอิงถึงตาราง auth_user

Table app_sightings

Column Name	Data type	Description
id	BIGINT	Primary Key ของข้อมูลการไปเยือน
arrivaldate	DATE	วันที่ที่ไปเยือน
created_at	TIMESTAMP	วันที่และเวลาที่สร้างข้อมูล
celebrities_id	BIGINT	Foreign Key ของสัตว์เลี้ยงที่ไปเยือน
places_id	BIGINT	Foreign Key ของสถานที่ที่พบเห็น
addby_auth_user_id	INTEGER	รหัสผู้ใช้ที่เพิ่มข้อมูลการพบเห็นนี้

Table app_comments

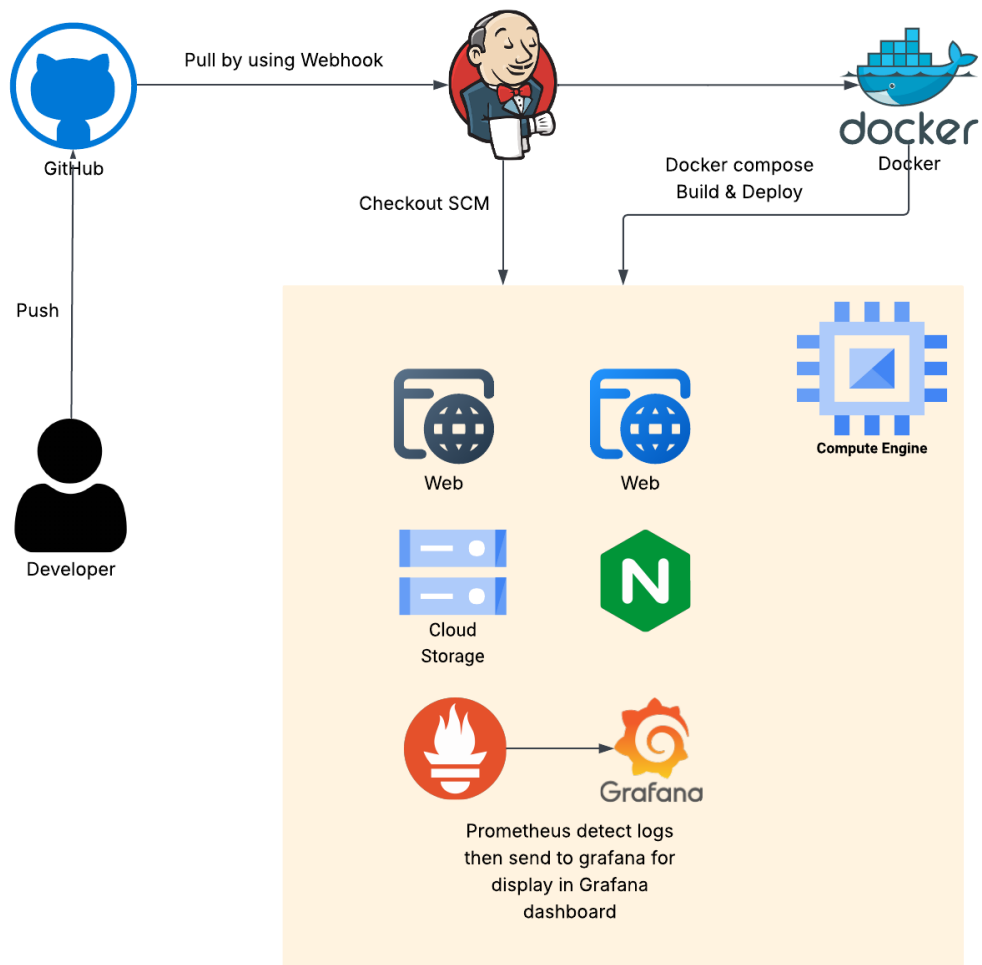
Column Name	Data type	Description
id	BIGINT	Primary Key ของความคิดเห็น
rating	SMALLINT	คะแนนรีวิวสถานที่
comment_text	TEXT	เนื้อหาความคิดเห็น
created_at	TIMESTAMP	วันที่และเวลาที่สร้างความคิดเห็น
updated_at	TIMESTAMP	วันที่และเวลาที่แก้ไขความคิดเห็น
places_id	BIGINT	Foreign Key ของสถานที่ที่ถูกแสดงความคิดเห็น
user_id	INTEGER	Foreign Key ของผู้ใช้ที่แสดงความคิดเห็น

Table auth_user

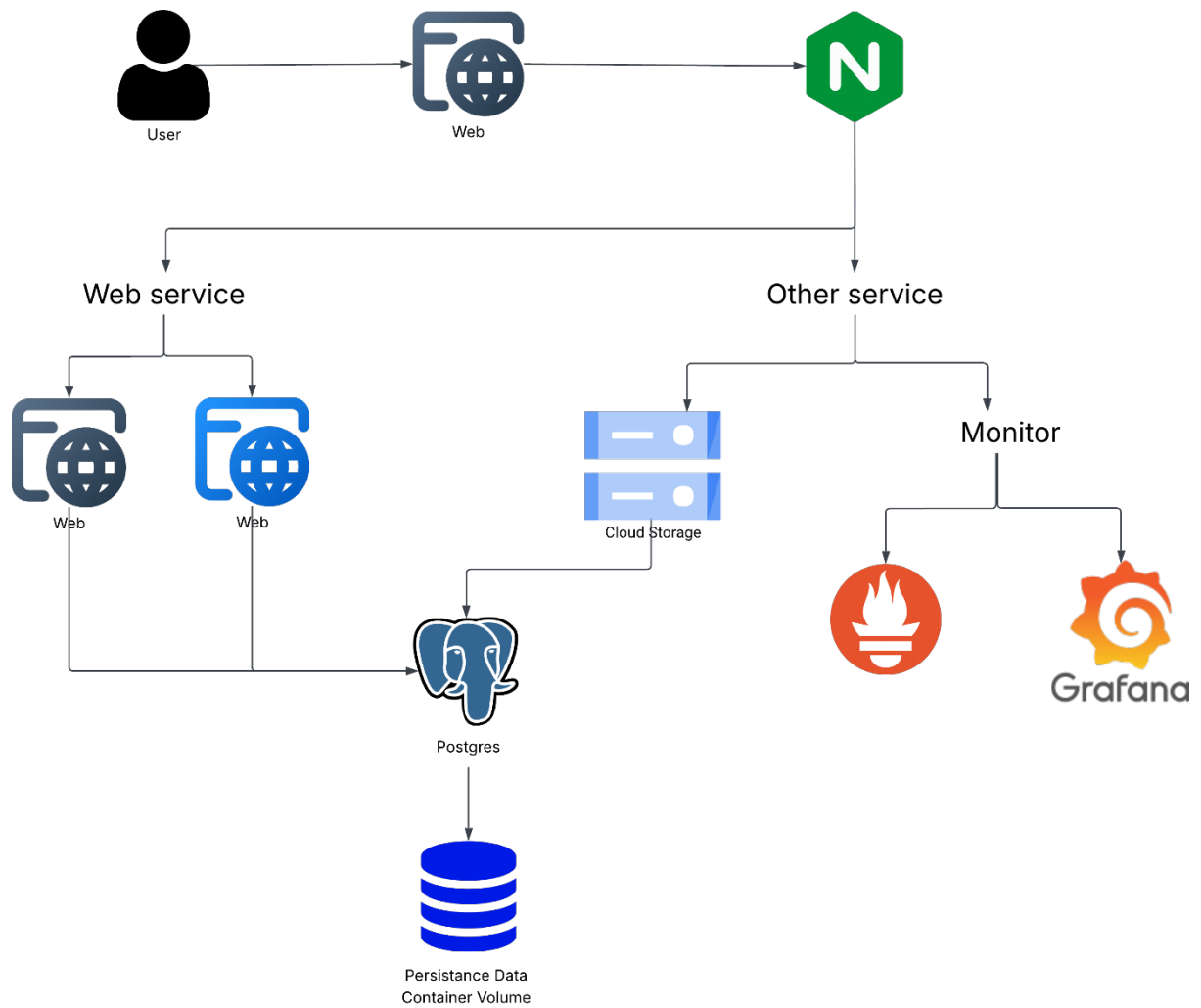
Column Name	Data type	Description
id	INTEGER	Primary Key ของผู้ใช้งาน
username	VARCHAR(150)	ชื่อผู้ใช้สำหรับเข้าระบบ
password	VARCHAR(128)	รหัสผ่านของผู้ใช้ (ถูกเข้ารหัส)
first_name	VARCHAR(150)	ชื่อจริงของผู้ใช้
last_name	VARCHAR(150)	นามสกุลของผู้ใช้
email	VARCHAR(254)	อีเมลของผู้ใช้
is_superuser	BOOLEAN	สถานะความเป็นผู้ดูแลระบบสูงสุด
is_staff	BOOLEAN	สถานะการเป็นทีมงาน
is_active	BOOLEAN	สถานะการเปิดใช้งานบัญชี
last_login	TIMESTAMP	วันที่และเวลาที่เข้าระบบล่าสุด
date_joined	TIMESTAMP	วันที่และเวลาที่สมัครใช้งาน

3.4 หลักการทำงานของระบบ

3.4.1 CI/CD Architecture



3.4.2 Load Balancing Architecture

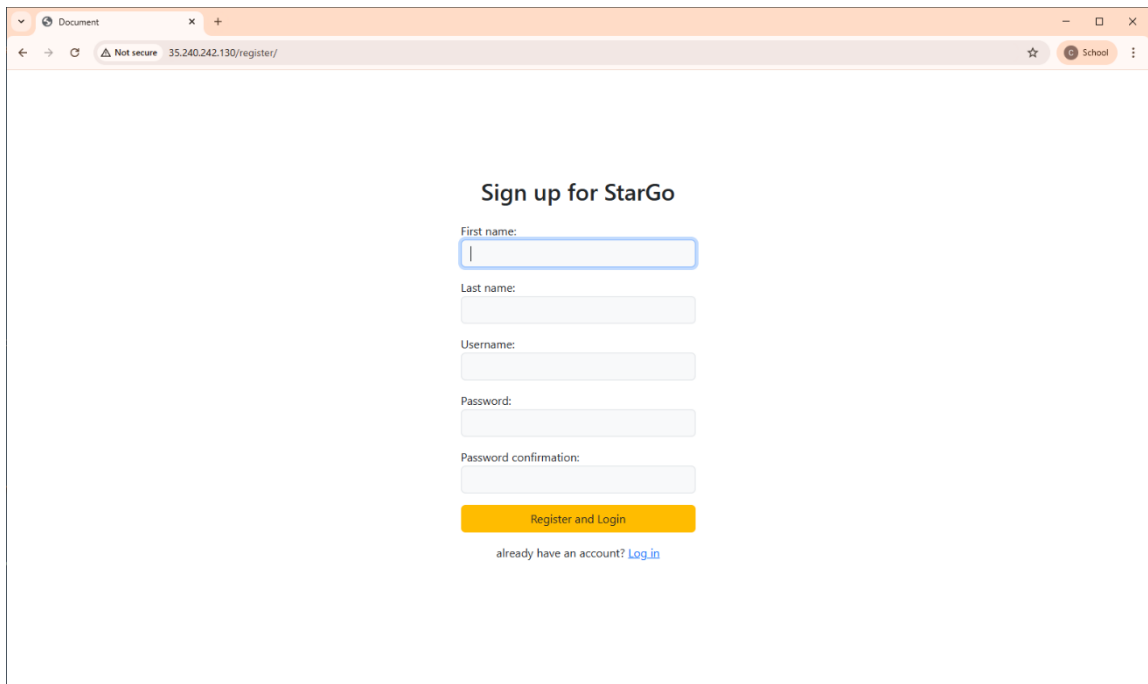


บทที่ 4

ผลการดำเนินงาน

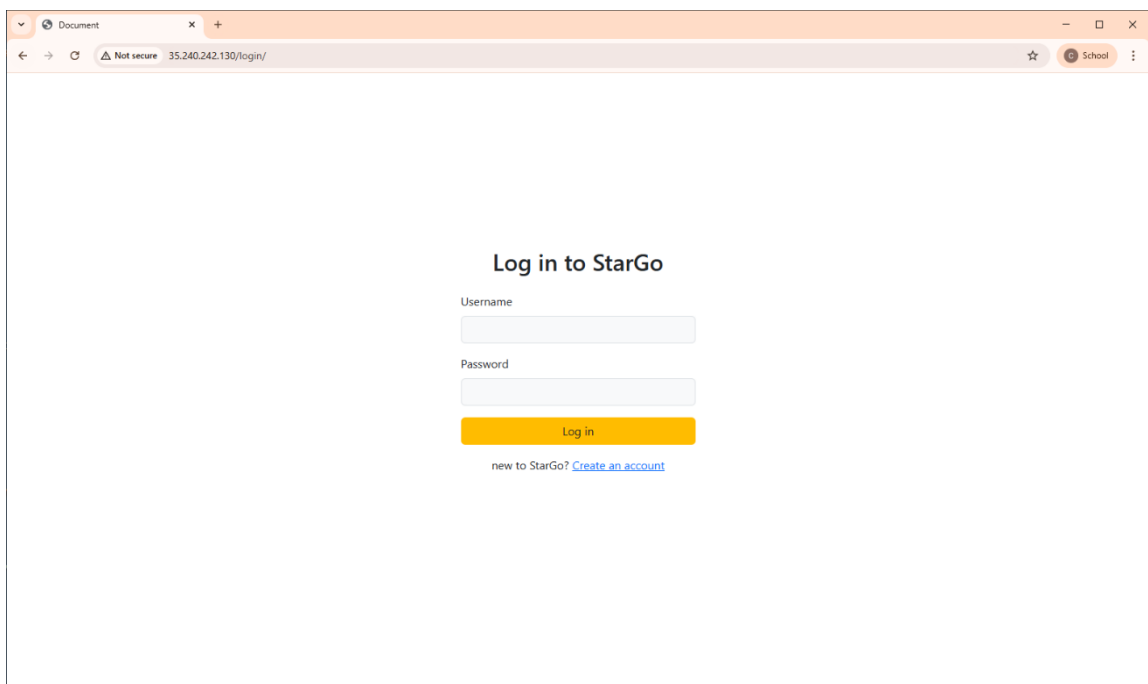
โครงการ PetGo จัดทำขึ้นเพื่อศึกษาและพัฒนาเว็บไซต์โดยใช้ระบบคลาวด์ จากความรู้วิชา Software Development Tools and Environments โดยมีผลการดำเนินงาน ดังนี้

หน้า ลงทะเบียน



A screenshot of a web browser showing the registration page for StarGo. The browser's address bar displays "35.240.242.130/register/". The page has a white background with a centered registration form. The form includes fields for "First name:", "Last name:", "Username:", "Password:", and "Password confirmation:". Below these fields is a yellow button labeled "Register and Login". At the bottom of the form, there is a link that says "already have an account? [Log in](#)".

หน้า เข้าสู่ระบบ



A screenshot of a web browser showing the login page for StarGo. The browser's address bar displays "35.240.242.130/login/". The page has a white background with a centered login form. The form includes fields for "Username" and "Password". Below these fields is a yellow button labeled "Log in". At the bottom of the form, there is a link that says "new to StarGo? [Create an account](#)".

หน้า ค้นหาสถานที่

StarGO Places Pets

Search Places

Place not found? [add here](#)

หน้า เพิ่มสถานที่

StarGO Places Pets


Add new place

Place name

Google Maps link

Address

Image



หน้า ลบสถานที่

Fashion Island | Change places | X

←

↻

Not secure 35.240.242.130/admin/app/places/17/change/

☆

📧

🔒

👤

⋮

🌈

Django administration

WELCOME ADMIN / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / App / Places / Fashion Island

Start typing to filter...

APP

Celebritiess

+ Add

Places

+ Add

Sightings

+ Add

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Users

+ Add

Change places

Fashion Island

HISTORY

Name:

Fashion Island

Googlemaplink:

https://maps.app.goo.gl/cR1F8f12LhST3GCHG

Address:

587-589/7-9 อ. รามอินทรา แขวงคันนายาว เขตคัน

Addby auth user:

testuser3

✎

+

👤

Imageurl:

Currently: http://storage.googleapis.com/images/c1507e3-694f-4364-90c9-94c30596b0f1.jpg

🗑

 Clear

Change:

Choose File

 No file chosen

SAVE

Save and add another

Save and continue editing

Delete

หน้า ค้นหาสัตว์เลี้ยง

Document X

←

↻

Not secure 35.240.242.130/stars/

🔍

☆

📧

🔒

👤

⋮

🌈

StarG@

Places

Pets

👤

Search pet stars

🔍

Stars not found? [add here](#)

หน้า เพิ่มรายชื่อสัตว์เลี้ยง

Document

35.240.242.130/stars/addnewstar/

StarGO Places Pets

Add Stars

Nickname

☐ Your pet?

Image

Choose File

No file chosen

Confirm

หน้า ลบสัตว์เลี้ยง

chihua | Change celebrities | Django administration

35.240.242.130/admin/app/celebrities/13/change/

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

HomeAppCelebritesschihua

Start typing to filter...

APP

CelebritessAdd

PlacesAdd

SightingsAdd

AUTHENTICATION AND AUTHORIZATION

GroupsAdd

UsersAdd

Change celebrities

chihua

Nickname:

chihua

Imageurl:

Currently: http://storage001/media/images/f3d04ee2-896d-404c-9d79-744734c57bf.jpg

Change:

Choose File

No file chosen

Addby auth user:

max123

Owner:

max123

SAVE

Save and add another


Save and continue editing

Delete

หน้า แก้ไขสัตว์เลี้ยงของตัวเอง

Document x New Tab x +

← → ↻ Not secure 35.240.242.130/stars/13/edit/ ☆ School ⋮

StarGO Places Pets 

Edit chihua


Nickname:

Imageurl:
 No file chosen

หน้า จัดการโปรไฟล์

Edit Profile - StarGO x +

← → ↻ Not secure 35.240.242.130/profile/edit/ ☆ School ⋮

StarGO Places Pets 

Edit Profile

Change Password


Delete Account

Edit Profile

First name

Last name

Username



[Upload a photo](#)
[Remove photo](#)

หน้า เปลี่ยนรหัสผ่าน

StarGO Places Pets

Edit Profile

Change Password

Delete Account

Change Password

Old Password

New Password

Confirm New Password

Confirm

หน้า ระบุการใช้งานผู้ใช้

StarGO Places Pets

Edit Profile

Change Password

Delete Account

Delete account

are you sure you want to delete account?

Confirm delete account

บทที่ 5

สรุปผลการดำเนินงาน

สรุปผลการดำเนินการ

จากการดำเนินโครงการในครั้งนี้ คณะผู้จัดทำสามารถดำเนินงานได้ตามวัตถุประสงค์ที่กำหนดไว้ โครงการได้ช่วยให้ผู้จัดทำเข้าใจและสามารถนำความรู้ที่ได้รับจากการเรียนมาประยุกต์ใช้ในทางปฏิบัติได้จริง โดยเฉพาะในส่วนของ การออกแบบระบบแบบ Microservices และการ Deploy ระบบ ให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพผ่านเครื่องมือ เช่น Docker, Jenkins สำหรับการทำให้ CI/CD, Grafana สำหรับการติดตามและแสดงผลการทำงานของระบบ นอกจากนี้ยังได้ฝึกทักษะการแก้ไขปัญหาที่พบระหว่างการพัฒนา ผลลัพธ์ของโครงการเป็นไปตามที่ตั้งเป้าไว้ และสามารถนำแนวทางการพัฒนาโครงการนี้ไปต่อยอดหรือปรับใช้ในโครงการอื่น ๆ ได้ในอนาคต

ปัญหาและอุปสรรคที่พบ

มีปัญหาในการร่วมกันระหว่าง Django กับ Nginx และ Django ทำให้ Microservice มีปัญหาบ่อย แต่ปัญหาทั้งหมดนี้ยังสามารถแก้ไขได้