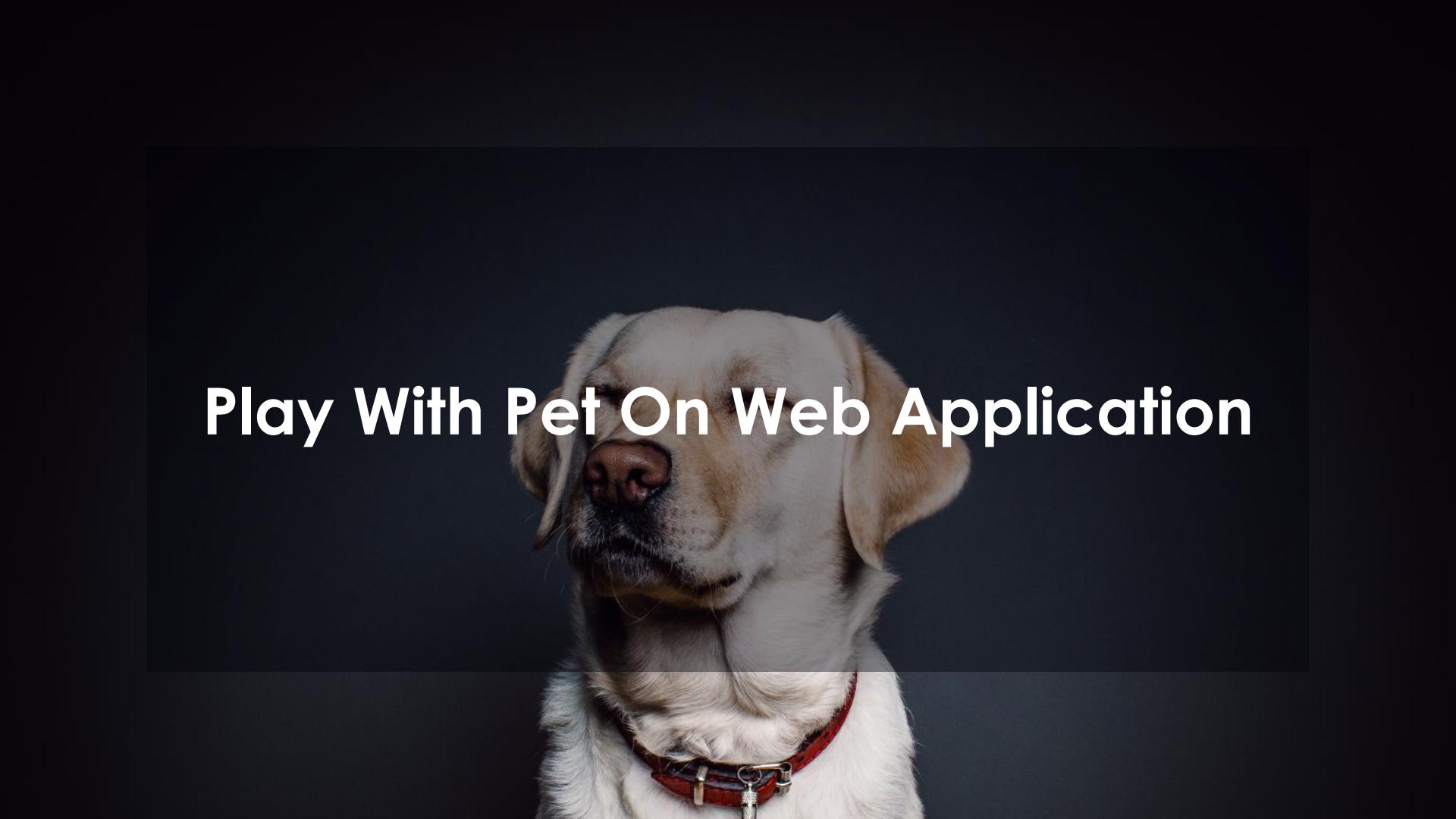


Mini Project

A close-up photograph of a light-colored dog, possibly a Golden Retriever or similar breed, looking slightly to the left. The dog is wearing a red leather collar with a silver buckle. The background is a solid, dark gray.

Play With Pet On Web Application



Submitted by

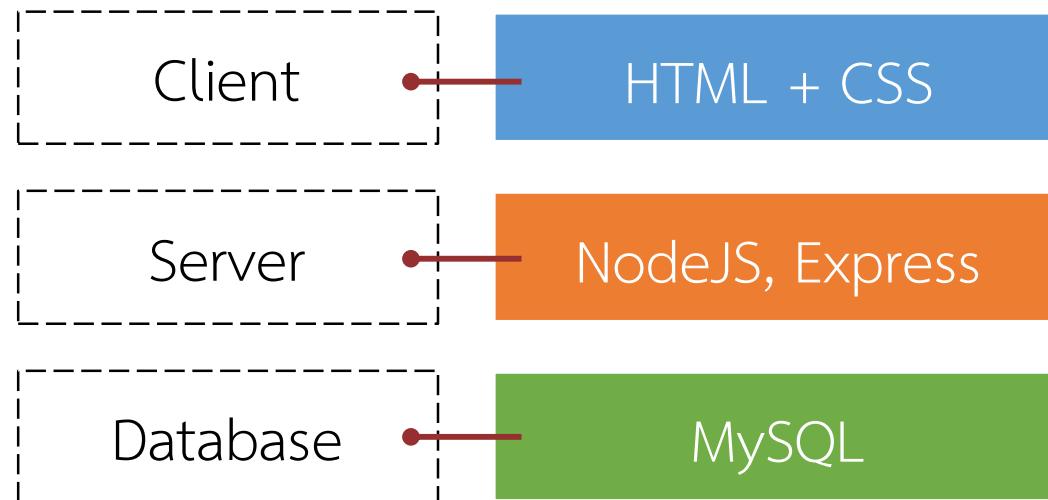
5701012620039 ສຸກາກຣນີ ບູແກ້ວ

5701012620195 ແບຮຮງາ ບຸກູຮີຮັກເຈີ

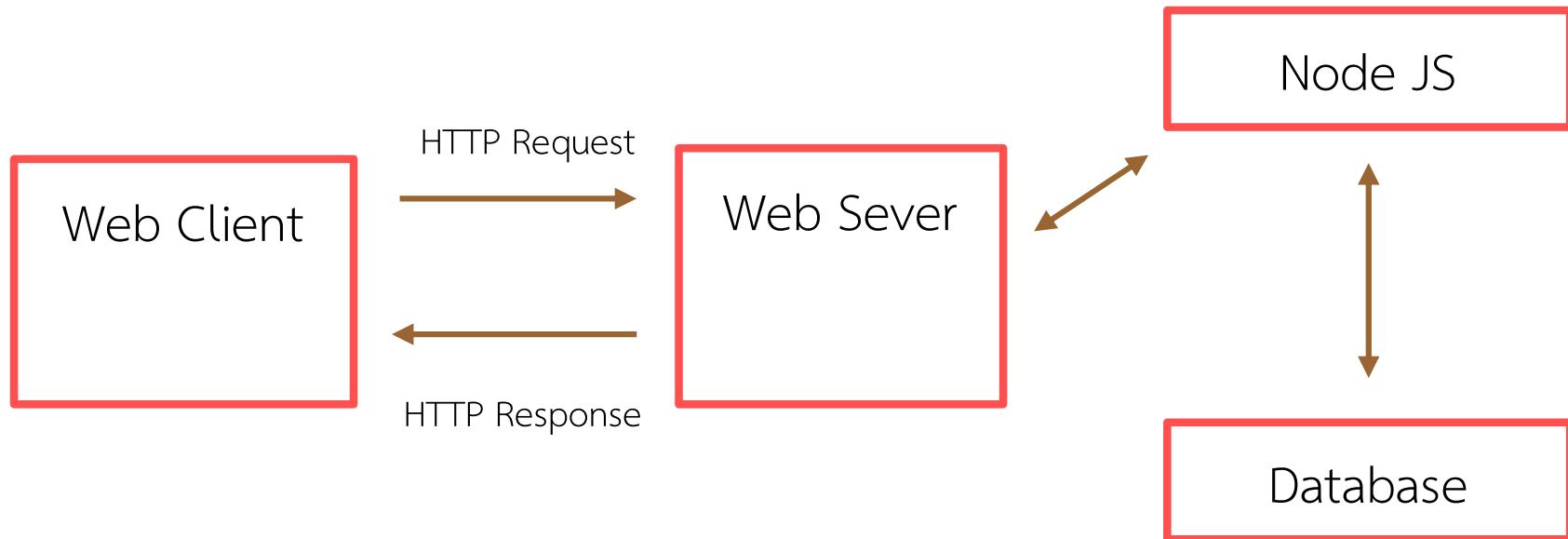


หลักการของระบบ

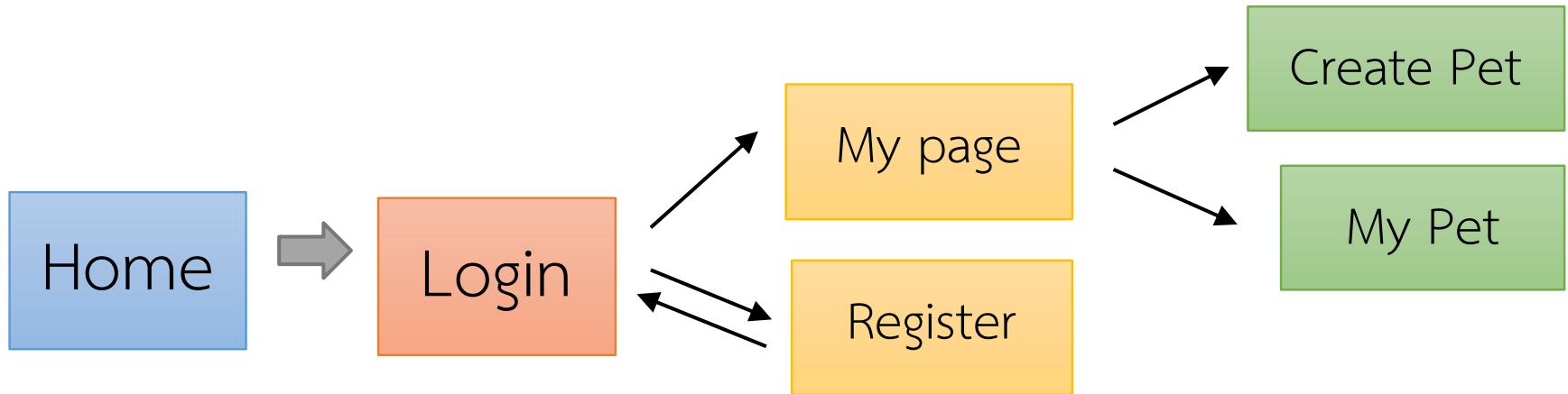
Web Application



การรับส่งข้อมูลจาก Web Server ไปยัง Web Client



Block Diagram สำหรับการทำงานของเว็บไซต์ Play With Pet





ขั้นตอนในการศึกษา

1. สร้างเว็บไซต์สำหรับการเล่นกับสัตว์เลี้ยง
2. สร้างระบบฐานข้อมูลเก็บข้อมูลของผู้ใช้งาน
3. ตรวจสอบการทำงานและปรับปรุงระบบ

1. สร้างเว็บไซต์สำหรับการเล่นกับสัตว์เลี้ยง

1.1 หน้าหลักของเว็บไซต์

1.2 หน้าต่างเข้าสู่ระบบ

- ช่องกรอกข้อมูล username, password

- ปุ่มเข้าสู่ระบบ

1.3 หน้าต่างสมัครเป็นผู้ใช้งาน

- ช่องกรอกข้อมูล username, password

- เลือกรหัส male หรือ woman

- ปุ่มลงทะเบียนสมัครเข้าใช้งาน

1.4 หน้าต่างสร้างสัตว์เลี้ยง

- ตั้งชื่อให้กับสัตว์เลี้ยง

- เลือกสัตว์เลี้ยงที่จะเล่นด้วยได้ คือ แมว สุนัข และหนูแฮมเตอร์

- มี level แสดงค่าประสบการณ์ของสัตว์เลี้ยง level และค่าประสบการณ์จะเพิ่มขึ้นเมื่อมีการให้อาหารหรือเล่นกับสัตว์เลี้ยง

1.5 หน้าต่างข้อมูลของผู้ใช้

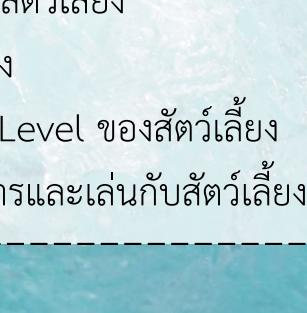
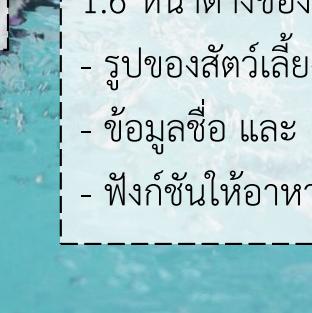
- ประกอบด้วยข้อมูลของสัตว์เลี้ยงที่ได้สร้าง

1.6 หน้าต่างของสัตว์เลี้ยง

- รูปของสัตว์เลี้ยง

- ข้อมูลชื่อ และ Level ของสัตว์เลี้ยง

- ฟังก์ชันให้อาหารและเล่นกับสัตว์เลี้ยง



2. สร้างระบบฐานข้อมูลเก็บข้อมูลของผู้ใช้งาน

- 2.1 ชื่อของผู้ใช้งาน (username)
- 2.2 รหัสผ่านของผู้ใช้งาน (password)
- 2.3 เพศของผู้ใช้งาน (gender)
- 2.4 ชื่อของสัตว์เลี้ยงของผู้ใช้งาน (pet name)
- 2.5 ชนิดสัตว์เลี้ยงของผู้ใช้งาน (pet type)
- 2.6 ค่าประสบการณ์ของสัตว์เลี้ยงของผู้ใช้งาน (exp)

3. ตรวจสอบการทำงานและปรับปรุงระบบ

วางแผนการทํางาน



สัปดาห์ที่	งาน	ระยะเวลา	ผู้รับผิดชอบ
1	คิดหัวข้อ และภาพรวมของระบบ	20 มี.ค. – 26 มี.ค. 2560	สุภารณ์ เพชรภูวा
2	- ออกรูปแบบเว็บไซต์ html	27 มี.ค. – 2 เม.ย. 2560	สุภารณ์
3	- Photoshop ออกรูปสัตว์เลี้ยง	3 เม.ย. – 9 เม.ย. 2560	สุภารณ์
4	- ควบคุมการทำงานด้วย JavaScript	10 เม.ย. – 16 เม.ย. 2560	สุภารณ์ เพชรภูว่า
5	- ระบบติดต่อฐานข้อมูล MySQL	17 เม.ย. – 23 เม.ย. 2560	เพชรภูว่า
6	- ทดสอบ NodeJS Express	24 เม.ย. – 30 เม.ย. 2560	เพชรภูว่า
7	- ทดสอบ NodeJS Express - ระบบติดต่อฐานข้อมูล MySQL	1 พ.ค. – 14 พ.ค. 2560	สุภารณ์ เพชรภูว่า
8	- ทดสอบระบบ ตรวจดูความถูกต้อง และแก้ไขงาน - จัดเตรียมเอกสารนำเสนอ	หลังสอบปลายภาค	สุภารณ์ เพชรภูว่า



Node JS



NodeJS เป็น Runtime ของ JavaScript ทำให้ระบบ Web server สามารถทำงานได้ รวมถึงจัดการไฟล์ในระดับระบบปฏิบัติการได้ เมื่อ用ภาษาโปรแกรมที่ร่วม



Express

Express เป็นตัวควบคุมจัดการกับฝั่ง Server เป็น Framework ของการเขียน Application ด้วย Node.js โดยใช้ภาษา JavaScript

- จัดการ Web Request (Request Router)
- จัดการ Cookie (Cookie Management)
- จัดการ Session (Session Management)



Database

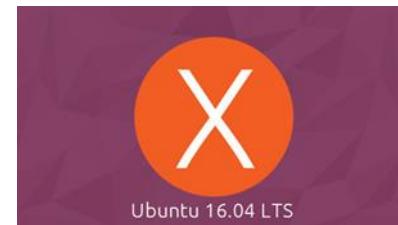


MySQL คือระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL เป็นซอฟต์แวร์โอเพนซอร์ส โดย sql เป็นภาษาที่ใช้ในสำหรับการเรียกใช้ฐานข้อมูล ซึ่งย่อมาจาก "structured query language" โดยที่ sql เป็นภาษาที่มีมาตรฐานและเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง sql กับฐานข้อมูลที่ชนิดใดก็ได้ และคำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดยึดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง



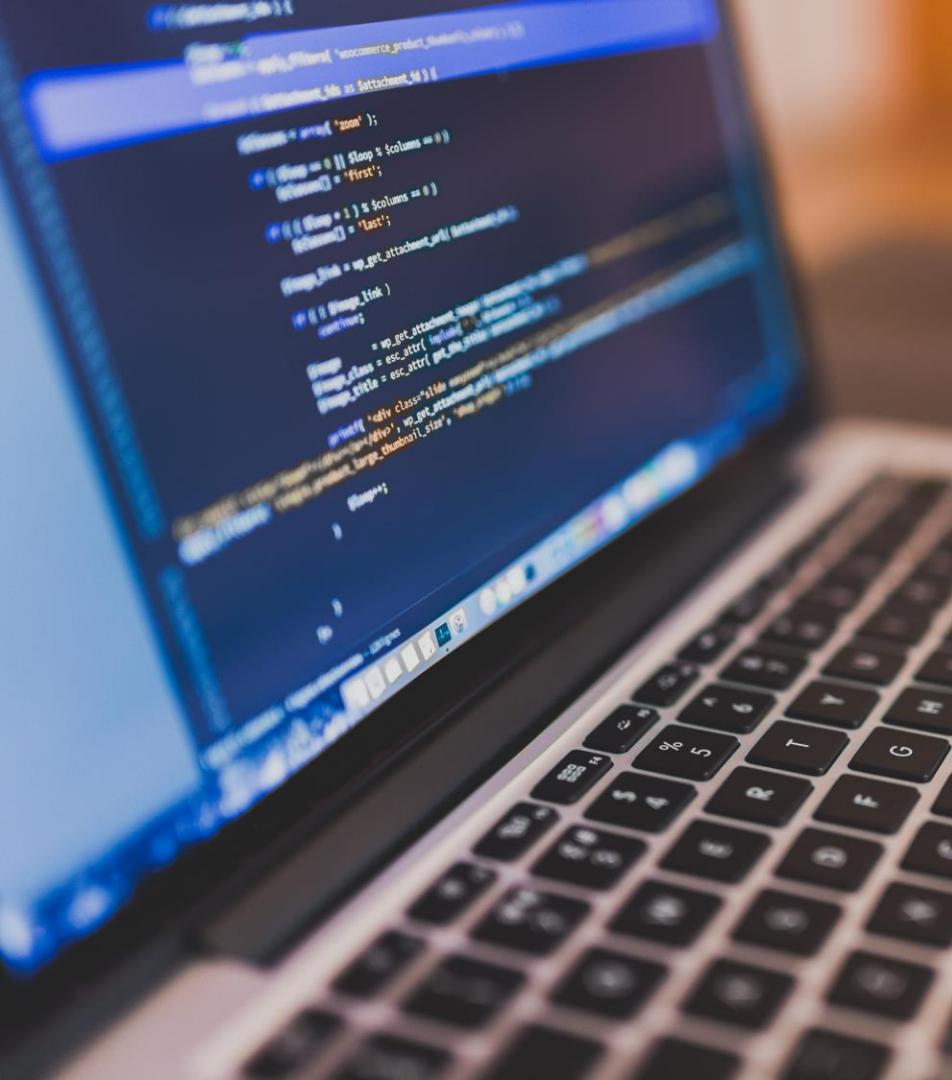
ระบบปฏิบัติการที่ใช้

Ubuntu 16.04



Windows 10



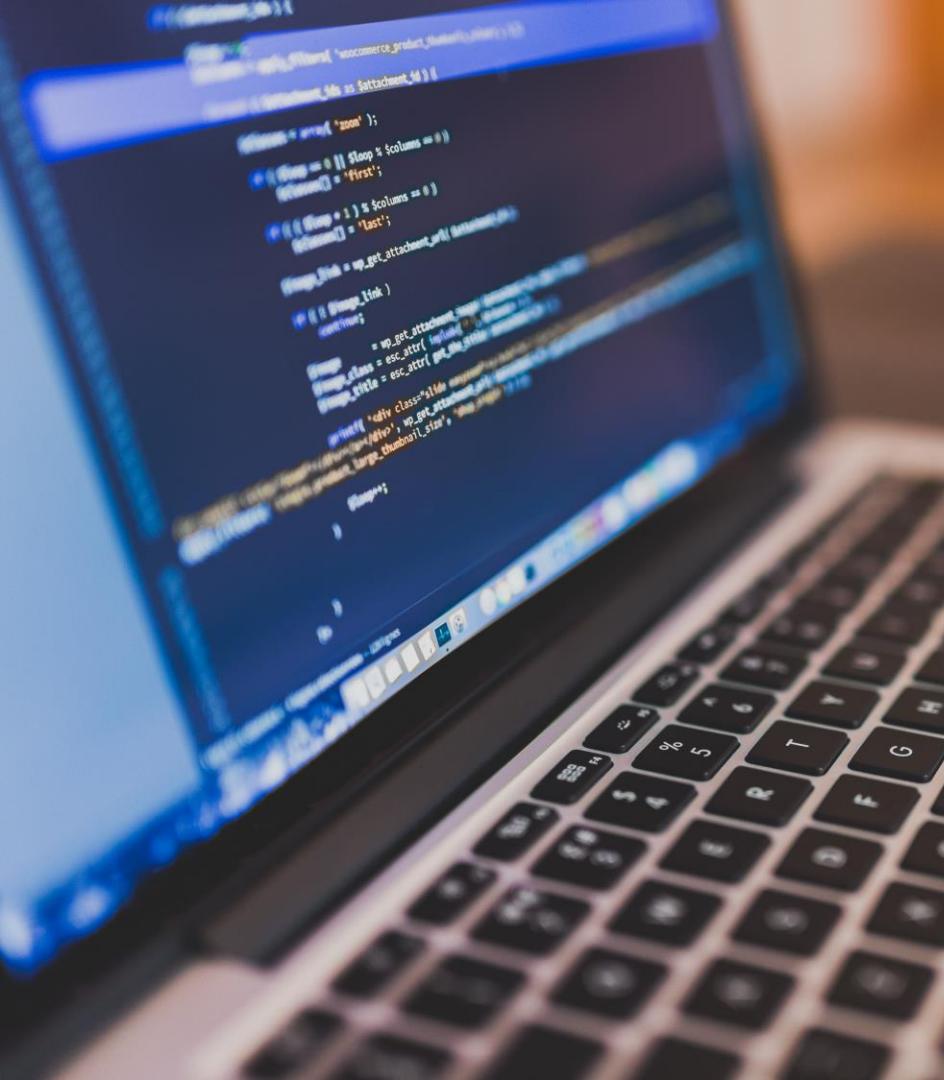


Software กี่ใช้

Komodo IDE : Version 10.0.1



Komodo IDE เป็นโปรแกรม Text editor ที่สามารถรองรับการใช้งานของโค้ดได้หลากหลายภาษา มากไม่ว่าจะเป็น HTML, PHP, JavaScript, CSS, XML, Python, Ruby ,TCL ช่วยเขียนโปรแกรมสำหรับ Developer ที่มีความหลากหลายมาก แก้ไข Text ไฟล์ต่าง ๆ ได้ มีฟังก์ชันเครื่องมือที่สร้างมาให้ใช้งานกับการเขียนโค้ด ได้เป็นอย่างดี



Software กี่ใช้

Photoshop CS6 และ Paint tool SAI



Photoshop เป็นโปรแกรมในตระกูล
Adobe สำหรับสร้างและแก้ไขรูปภาพ

ส่วนโปรแกรม Paint tool SAI ของ
ผู้พัฒนาจากญี่ปุ่น เป็นโปรแกรมที่ช่วยในการวาดภาพ

```
style.css
427 color: rgba(55, 55, 55, 0.5);
428 font-weight: normal;
429 opacity: 1;
430 }
431
432 *:-ms-input-placeholder {
433 color: rgba(55, 55, 55, 0.5);
434 font-weight: normal;
435 opacity: 1;
436 }
437
438 /* Buttons
439 ===== */
440 .btn, .comment-reply-link,
441 input[type="button"],
442 input[type="submit"],
443 input[type="reset"],
444 button {
445 display: inline-block;
446 font-weight: normal;
447 margin: 0;
448 line-height: normal;
449 border: none;
450 box-shadow: none;
451 text-align: center;
452 vertical-align: middle;
453 cursor: pointer;
454 white-space: nowrap;
455 transition: 0.1s background-color ease, color 0.1s ease, 0.1s border-color ease;
456 -webkit-user-select: none;
457 -moz-user-select: none;
458 user-select: none;
459 border-radius: 0;
460 width: auto;
461 height: auto;
462 background-image: none;
463 padding: 11px 26px;
464 background-color: #58c5d0;
465 color: #fff;
466 font-size: 13px;
467 }
468 .btn:active, .comment-reply-link:active,
469 input[type="button"]:active,
470 input[type="submit"]:active,
471 input[type="reset"]:active,
472 button:active {
473 outline: none;
474 }
475 .btn:hover, .comment-reply-link:hover,
476 input[type="button"]:hover,
477 input[type="submit"]:hover,
478 input[type="reset"]:hover,
479 button:hover {
480 background: #2a7280;
481 color: #fff;
482 text-decoration: none;
483 }
484 .btn:focus, .comment-reply-link:focus,
485 input[type="button"]:focus,
486 input[type="submit"]:focus,
487 input[type="reset"]:focus,
488 button:focus
```

```
main.css
1 * html {
2 background: Helvetica;
3 background-image: linear-gradient(270deg, #f0f0f0, #e0e0e0, #d0d0d0);
4 -webkit-font-smoothing: antialiased;
5 }
6
7 * body {
8 background: #f0f0f0;
9 background: #f0f0f0 0 0px rgba(0, 0, 0, 0.05);
10 color: #555555;
11 font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
12 font-size: 14px;
13 line-height: 1.3;
14 margin: 0 auto;
15 max-width: 800px;
16 padding: 2em 0 0 0;
17 }
18
19 * h1, h2, h3, h4, h5, h6 {
20 color: #222222;
21 font-weight: bold;
22 line-height: 1.1;
23 }
24
25 * h1 {
26 margin-top: 1.3em;
27 }
28
29 * a {
30 color: #008080ff;
31 }
32
33 * b, strong {
34 font-weight: bold;
35 }
36
37 * samp {
38 display: none;
39 }
40
41 * img {
42 -webkit-animation: cubic-bezier(0, 0, 0, 1) 1s;
43 animation: cubic-bezier(0, 0, 0, 1) 1s;
44 background: transparent;
45 border: 1px solid #ccc;
46 border-radius: 50%;
47 display: block;
48 margin: 1.5em auto;
49 max-width: 100px;
50 }
51
52 * @-webkit-keyframes colorize {
53 0% {
54 color: #008080ff;
55 }
56 100% {
57 color: #008080ff;
58 }
59 }
60
61 * @keyframes colorize {
```

ภาษาที่ใช้

JavaScript

HTML5

CSS3

JavaScript



JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต เป็นภาษาสคริปต์เชิงวัตถุที่เรียกว่า script ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ แปลความและดำเนินงานไปทีละคำสั่ง (interpret) หรือเรียกว่า Object Oriented Programming ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่ง Client และ ทางฝั่ง Server

HTML5



HTML ย่อมาจากคำว่า Hypertext Markup Language คือ ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างกัน เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink HTML5 เป็นภาษาที่ได้รับการพัฒนาขึ้นมาใหม่โดยจะมี Tag ใหม่ๆ เพิ่มเติมขึ้นมาเพื่อให้สอดคล้องกับการใช้งานมากยิ่งขึ้น

- Doctype เขียนง่ายขึ้น
- การกำหนดภาษาทำได้ง่ายขึ้น
- การกำหนดชุดตัวอักษรทำได้ง่ายขึ้น
- ไม่ต้องมี “/” สำหรับ tag เดียวแล้ว (tag ที่ไม่มี tag ปิด)
- Tag บางส่วนจะไม่รองรับใน HTML5 แล้ว

ความสามารถของ HTML5

- Semantic Markup: โค้ดเป็นระเบียบทามให้ Search Engine เก็บข้อมูลได้ง่าย
- Form Enhancement: เพิ่มประสิทธิภาพของฟอร์ม
- Canvas: เอาไว้วาดรูป ตกแต่งรูป
- Content Editable: สามารถลิกกับข้อความในเว็บเพื่อแก้ไขจากตรงนั้นได้เลย
- Drag and Drop: ลากของมารวบ
- Persistent Data Storage: การเก็บข้อมูลบนเครื่องผู้ใช้ ซึ่งสามารถเก็บได้ถาวรสิบฐานข้อมูล



CSS ย่อมาจาก Cascading Style Sheets เป็นภาษาที่มีรูปแบบการเขียน Syntax ที่เฉพาะ และถูกกำหนดมาตรฐานโดย W3C (World Wide Web Consortium) เช่นเดียวกับ HTML และ XHTML ใช้สำหรับตกแต่งเอกสาร HTML/ XHTML ให้มี หน้าตา สีสัน ตัวอักษร เส้นขอบ พื้นหลัง ระยะห่าง ฯลฯ อย่างที่เราต้องการ ด้วยการ กำหนดคุณสมบัติให้กับ Element ต่าง ๆ ของ HTML เช่น <body>, <p>, <h1> เป็นต้น



1

การวิจัยและการสร้างรูปแบบ ของหน้าต่างเว็บไซต์

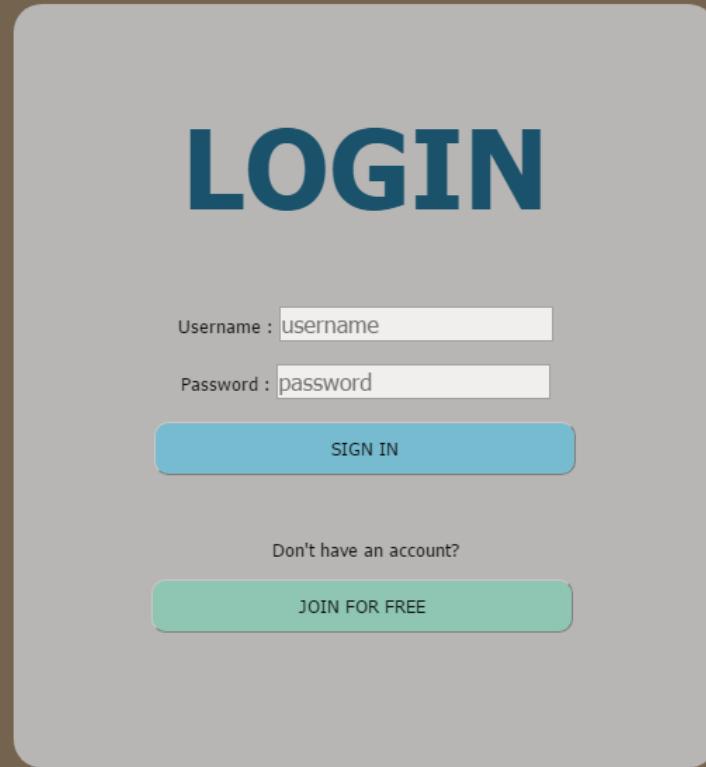
หน้าต่างเว็บไซต์

1. หน้าหลักของเว็บไซต์ (home.html)



เมื่อกดปุ่ม GET START แล้วจะเปลี่ยนไปยังหน้า Log in

2 หน้าต่างเข้าสู่ระบบ (login.html)



ส่วนของช่อง username และ password จะเป็นรูปแบบ text

กดปุ่ม sign in แล้วจะเข้าสู่หน้า page

กดปุ่ม sign up แล้วจะเข้าสู่หน้า register

3 หน้าต่างสมัครเป็นผู้ใช้งาน (register.html)

CREATE ACCOUNT

*Username :

*Password :

*Confirm Password :

Male Female

SIGN UP

Already have an account?

LOGIN

เมื่อกดปุ่ม จะส่งข้อมูลไปยัง Database และมีการตรวจสอบชื่อที่ซ้ำกัน สามารถแจ้งเตือนชื่อ Username ที่ซ้ำได้

4 หน้าต่างสร้างสัตว์เลี้ยง (create-pet.html)

X

CREATE PET

NAME PET :

SELECT PET :

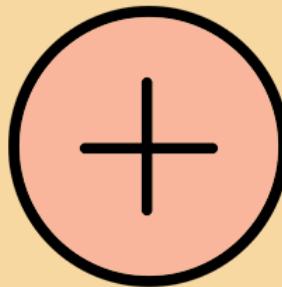
LET'S GO !



ในหน้า Create Pet มีให้กรอกชื่อข้อมูลของสัตว์เลี้ยง และเลือกชนิดของสัตว์เลี้ยงที่ต้องการ
กดปุ่ม “SUBMIT” เพื่อยืนยันข้อมูล และส่งข้อมูลไปยัง Database

5 หน้าต่างข้อมูลของผู้ใช้ (profile.html)

(Username) Logout



ในหน้า page จะแสดงข้อมูลชื่อของผู้ใช้ และข้อมูลของสัตว์เลี้ยง
เมื่อคลิกที่รูปเครื่องหมาย + จะเข้าไปที่หน้า create pet

6 หน้าต่างของสัตว์เลี้ยง (pet.html)



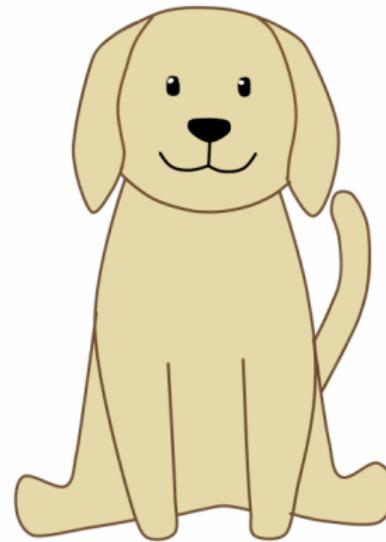
Play With Pet



FEED



PLAY



(Username) My Profile Logout

NAME : (Example)

LEVEL : 1

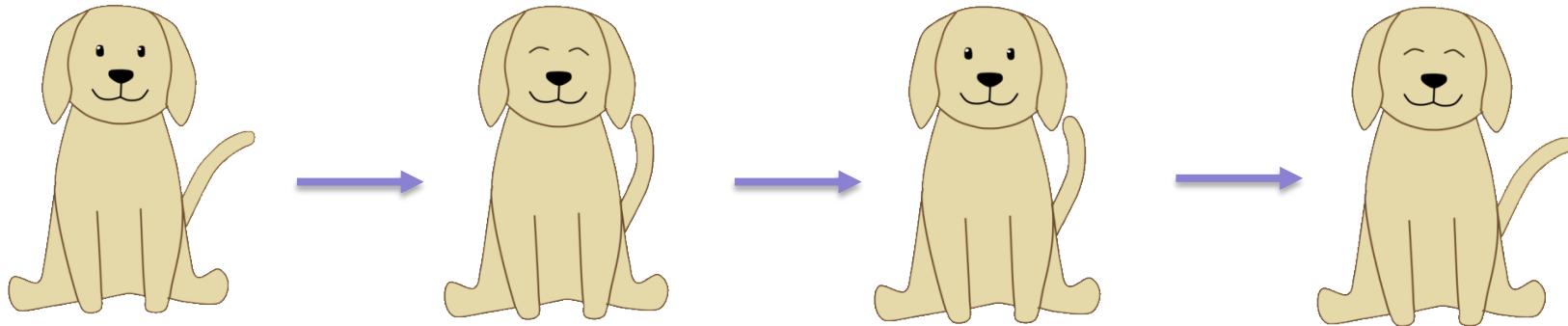
EXP : 

เมื่อกดที่รูป Feed หรือ Play จะโหลดรูปภาพกิจกรรมนั้น ๆ ที่ได้เลือก ในช่องที่แสดง Level นั้น exp จะขึ้นตามกิจกรรมที่ได้กดเลือกในแต่ละครั้ง เมื่อ exp ครบตามกำหนด Level ก็จะเพิ่มขึ้นไปตามลำดับ

Action ຕ່າງ ປົວ PET

ໃນຫຼາຕ່າງ PET

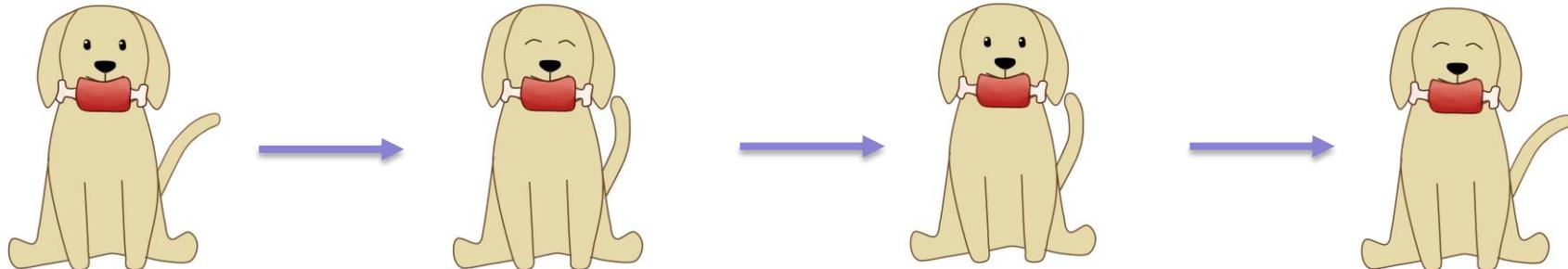
- ສ້າງເລື່ອງທອນປກທີ



ໃນຫຼາຕ່າງ PET

- ສ້າງເລື່ອງທອນໄດ້ຮັບອາຫາກ

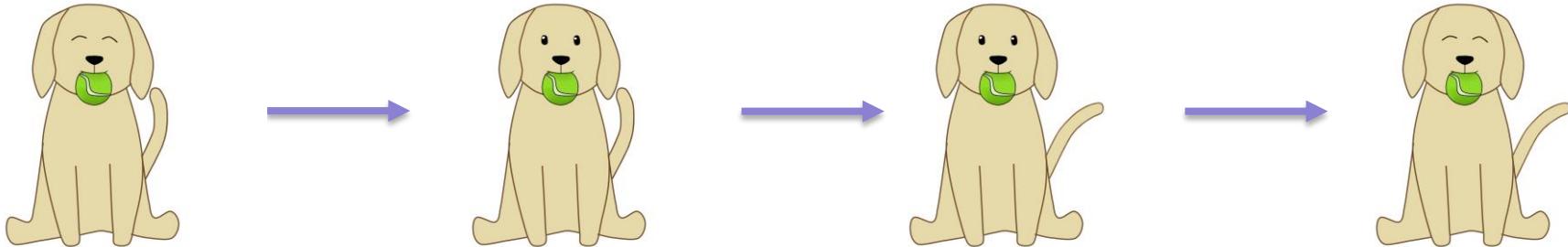
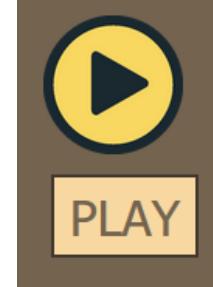
ເມື່ອກຳປຸ່ມ FEED ສ້າງເລື່ອງຈະໄດ້ຮັບອາຫາກ



ໃນຫຼາຕ່າງ PET

- ສ້າງເລື່ອງທອນກຳລັງເລີ່ມ

ເນື່ອກດປຸ່ມ PLAY ສ້າງເລື່ອງຈະໄດ້ຮັບຂອງເລີ່ມຫຼືກຳລັງເລີ່ມ



ໃນຫຼາຕ່າງ PET

- ມີສັດວ່າເລື່ອງທັງໝາດສາມະນິດໃຫ້ເລື່ອກ ນອກຈາກ Dog ແລ້ວຍັງມີ Cat ແລະ Hamster

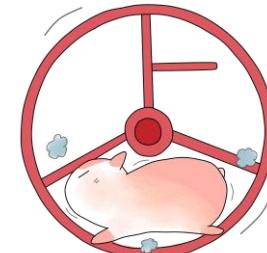
ຕອນປກຕີ



ຕອນໄດ້ຮັບອາຫາຣ



ຕອນເລັ່ນ



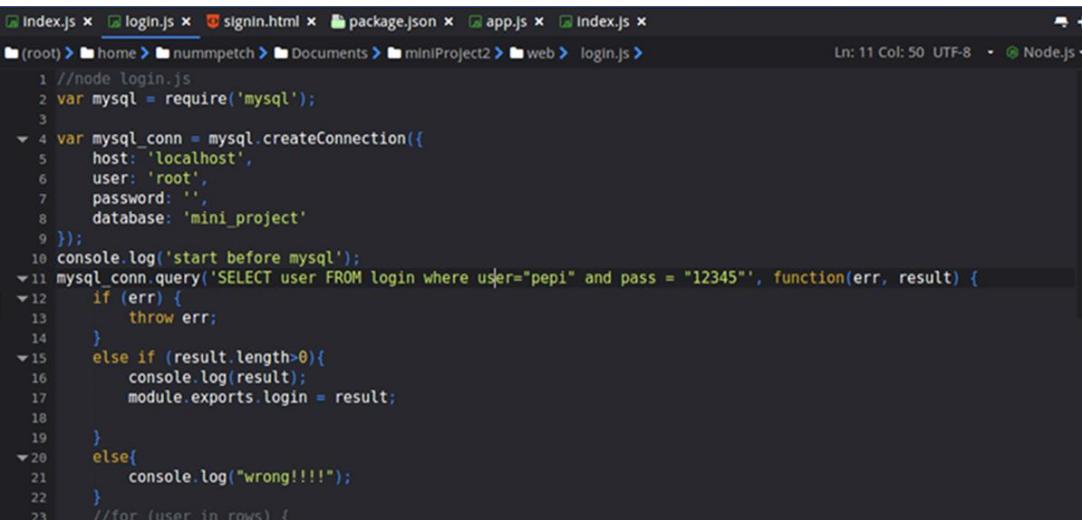


2

การใช้งาน NodeJS ติดต่อ กับฐานข้อมูล

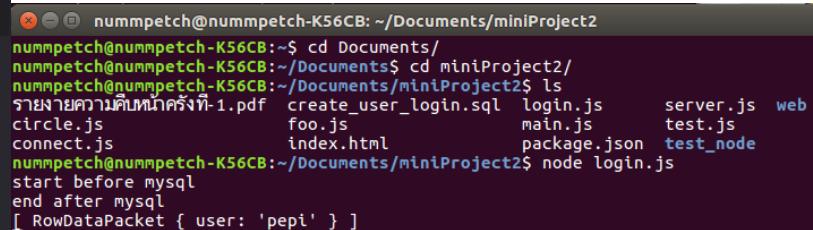
ทดสอบ NodeJS

ກណສວບການໃຊ້ງານ NodeJS ຕິດຕ່ອງກັບຈານຂ້າມູລ SQL



```
Index.js x login.js x signin.html x package.json x app.js x Index.js x
(root) > home > nummpetch > Documents > miniProject2 > web > login.js >
Ln: 11 Col: 50 UTF-8 Node.js

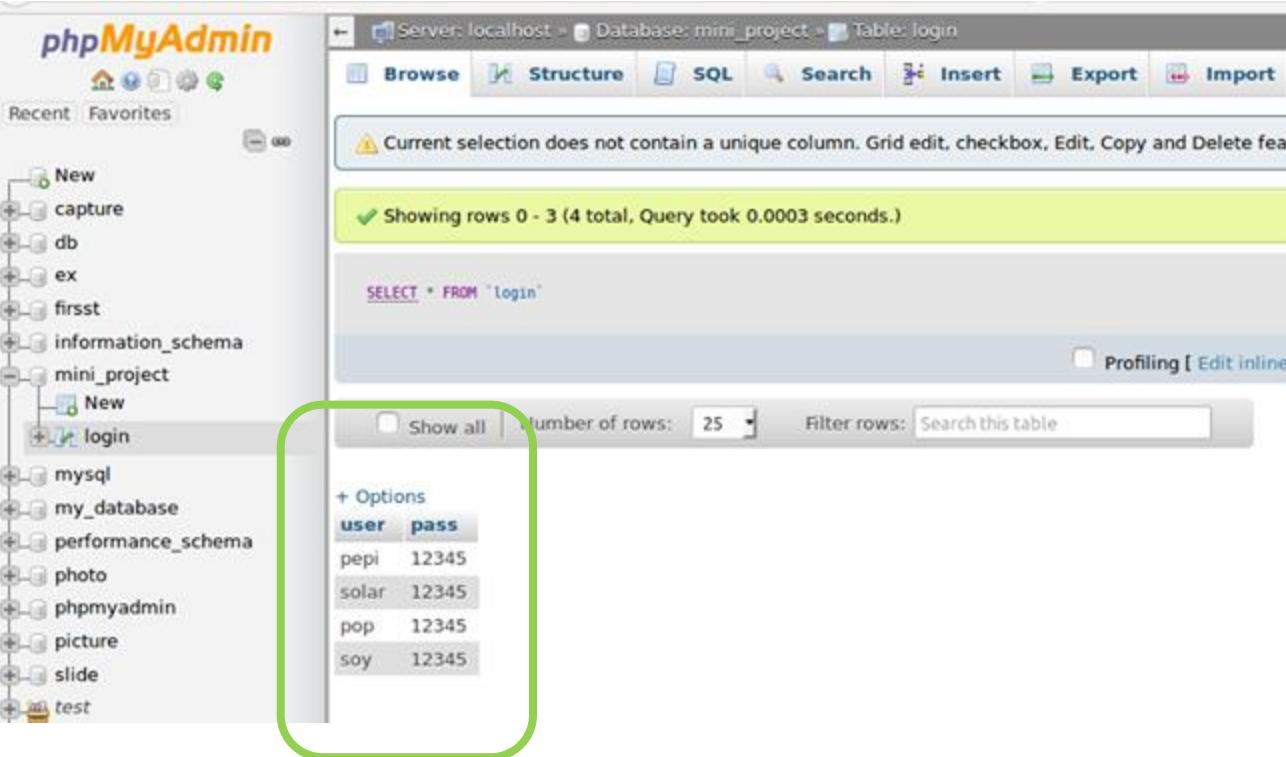
1 //node login.js
2 var mysql = require('mysql');
3
4 var mysql_conn = mysql.createConnection({
5   host: 'localhost',
6   user: 'root',
7   password: '',
8   database: 'mini_project'
9 });
10 console.log('start before mysql');
11 mysql_conn.query('SELECT user FROM login where user="pepi" and pass = "12345"', function(err, result) {
12   if (err) {
13     throw err;
14   }
15   else if (result.length>0){
16     console.log(result);
17     module.exports.login = result;
18   }
19   else{
20     console.log("wrong!!!!");
21   }
22   //for (user in rows) {
```



```
nummpetch@nummpetch-K56CB: ~/Documents/miniProject2
nummpetch@nummpetch-K56CB:~/Documents/miniProject2/
nummpetch@nummpetch-K56CB:~/Documents/miniProject2$ ls
รายงานความคืบหน้าครั้งที่-1.pdf  create_user_login.sql  login.js    server.js  web
circle.js                           foo.js        main.js    test.js
connect.js                         index.html   package.json  test_node
nummpetch@nummpetch-K56CB:~/Documents/miniProject2$ node login.js
start before mysql
end after mysql
[ RowDataPacket { user: 'pepi' } ]
```

ทดลองการใช้งาน NodeJS ติดต่อกับฐานข้อมูล SQL

กรณีที่ผู้ใช้งานเข้าสู่ระบบ เมื่อผู้ใช้งานป้อนข้อมูลในหน้าเว็บ sign in ข้อมูลจะถูกเก็บไว้ที่ Database



The screenshot shows the phpMyAdmin interface for a database named 'mini_project'. The left sidebar lists various databases and tables. The 'login' table under the 'mini_project' database is selected. The main area displays the table structure and data.

Table Structure:

+ Options	user	pass
	pepi	12345
	solar	12345
	pop	12345
	soy	12345

Data:

user	pass
pepi	12345
solar	12345
pop	12345
soy	12345

A green oval highlights the data grid containing the user names and their corresponding passwords.

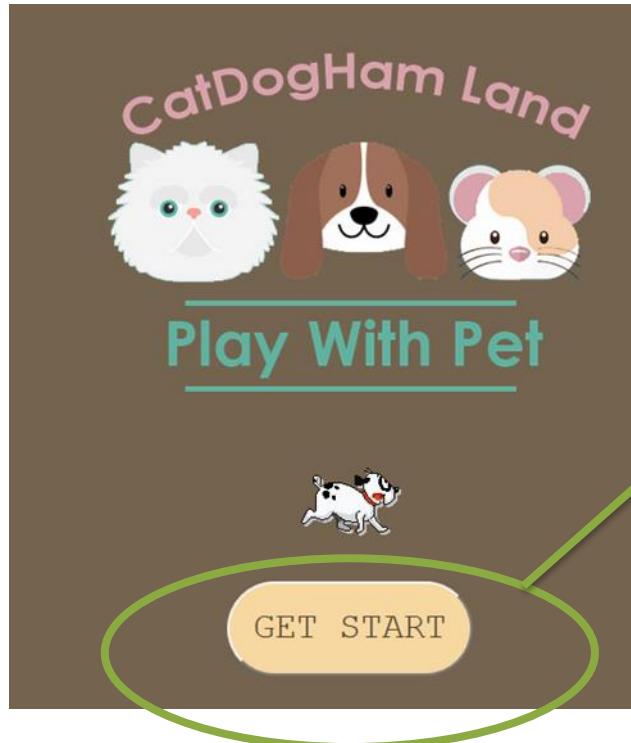


3

การใช้งานของ Web Application

Frist Page

หน้าแรกของเว็บไซต์
localhost:8080/



GET
localhost:8080/signin

ผู้ server :
get(signin)
load page/signin.html

Client Log in

เมื่อกดปุ่ม login จะทำการตรวจสอบว่ามีผู้ใช้งานที่ username และ password ตามที่ระบุหรือไม่



post :
localhost:8080
/login?name="..."&pass="..."

รอรับข้อมูล receive (data)
และ get

localhost:8080
table = "data"

Screenshot of a browser developer tools Network tab showing network requests:

- A successful GET request to `http://localhost:8080/data` with status `HTTP/1.1 200 OK` and duration `1ms`. The response body contains `signin:97:17`.
- An unsuccessful GET request to `http://localhost:8080/data` with status `HTTP/1.1 404 Not Found` and duration `4ms`. The response body contains `signin:97:17`.
- An unsuccessful GET request to `http://localhost:8080/data` with status `HTTP/1.1 404 Not Found` and duration `1ms`. The response body contains `signin:97:17`.

Client Log in

โดย function login() จะรับค่าข้อมูล username และ pass ของผู้ใช้งานเพื่อนำไปตรวจสอบ
เมื่อข้อมูลถูกต้องตรงตามตารางใน Database จะเข้าสู่หน้าถัดไป และเมื่อกรอกข้อมูลไม่ถูกต้อง จะขึ้นหน้าต่างข้อความว่า “wrong”

```
function login(){
    username = document.getElementById("username").value;
    pass = document.getElementById("password").value;
    send_data = 'name='+username+"&password="+pass;

    ajaxPost('/login?'+send_data, recieve_data);
    console.log("call");
}

function recieve_data(data){
    //console.log("data="+data);
    if(data.length>0){
        console.log("call pro="+data);
        window.location.href = "http://localhost:8080/getProfile?table="+data;
        // ajaxPost('/profile?table='+data);

    }
    else_if(data == ""){
        alert("wrong");
    }
}
```

Server Log in

function login() จะเก็บข้อมูลของผู้ใช้งานลงบน table ใน database

```
function login(name,pass){  
  
    connection.query('SELECT * FROM `user` WHERE name = "'+name+'" and pass="'+pass+'"', function(err, rows) {
```

```
//http://localhost:8080/login?name=pepi&password=123456  
app.post('/login?',function(req,res){  
    console.log("call post");  
    res.setHeader('Content-Type', 'application/json');  
    var getjson = getKeys(req.query);  
    //var key = getjson[0] ; //name,pass  
    var value = getjson[1];  
    var dataSplit = value.toString().split(',');  
    var name = dataSplit[0];  
    var pass = dataSplit[1];  
    //var data = getValuejson(req.query,key);  
    //console.log("key="+name);  
    //console.log("value="+pass);  
    login(name,pass);  
    res.send(json);  
    console.log("json="+json);  
});
```

↑
return name
↑ receive_data(data)

Server Log in

เมื่อกดปุ่ม sign up จะทำการตรวจสอบว่า password ตรงกันหรือไม่ หากตรงกันจะทำการ insert ผู้ใช้งานบน table

CREATE ACCOUNT

*Username :

*Password :

*Confirm Password :

Male Female

SIGN UP

Already have an account?

LOGIN



Table ของ user ใน Database MySQL

name	pass
pepi	123456
Tomson	123456
mama	123456
mama	123456
some	123456
some	123456
leo	123456
leo	123456

Client register

function regis

```
function regis(){
    var name = document.getElementById("username").value;
    var pass1 = document.getElementById("password1").value;
    var pass2 = document.getElementById("password2").value;
    if(pass1 == pass2){
        console.log("regis");
        ajaxPost('/regis?name=' + name + "&pass=" + pass1, recieve_data);
    }
}
```

```
function recieve_data(data){
    var name = document.getElementById("username").value;
    if(data == "true"){
        console.log("goto next");
        console.log(name);
        window.location.href = "http://localhost:8080/getProfile?table=" + name;
    }
}
```

ผู้ server :

Insert Into user name,pass

return true

Client register

```
function regis(name,pass){  
  connection.query('INSERT INTO `user`(`name`, `pass`) VALUES ("'+name+'", "'+pass+'")', function(err, rows) {  
    if (err) throw err;  
    console.log("Regis Success");  
  });  
}
```

```
app.post('/regis?',function(req,res){  
  console.log("call regis");  
  res.setHeader('Content-Type', 'application/json');  
  var getjson = getKeys(req.query);  
  //var key = getjson[0] ; //name,pass  
  var value = getjson[1];  
  var dataSplit = value.toString().split(',');  
  var name = dataSplit[0];  
  var pass = dataSplit[1];  
  //var data = getValuejson(req.query,key);  
  //console.log("key="+name);  
  console.log("name="+name);  
  console.log("pass="+pass);  
  regis(name,pass);  
  res.send(result_regis);  
  //console.log("json="+json);  
});
```

Table ของ user ใน Database MySQL

name	pass
pepi	123456
Tomson	123456
mama	123456
mama	123456
some	123456
some	123456
leo	123456
leo	123456

Client Create Pet

เมื่อกดปุ่ม Let's go จะ Insert ค่าลงในตารางว่า ได้ create pet type ตามที่ระบุเอาไว้ ในเวลาที่เท่าไหร่ และใครเป็นผู้สร้าง

Client Create Pet

function สร้างสัตว์เลี้ยง

```
function createPet(){
    var name = document.getElementById("name-pet").value;
    var e = document.getElementById("type");
    var type = e.options[e.selectedIndex].value;
    console.log(name+":"+type);
    ajaxPost('/createPet?name=' + name + '&type=' + type, recieve_data);
}

function recieve_data(data){
    console.log("namePet:" + data);
    if(data.length>0){
        window.location.href = "http://localhost:8080/Pet?data=" + data;
    }
}
```

Server Create Pet

```
'''  
function createPet(namePet,type){  
    var text = 'INSERT INTO `pet`(`user`, `pet_name`, `type`, `exp`, `time`, `action`) VALUES ('+namepath+",''+namePet+'","'+type+'",'+''1'+,CURRENT_TIMESTAMP,"create pet")';  
}
```

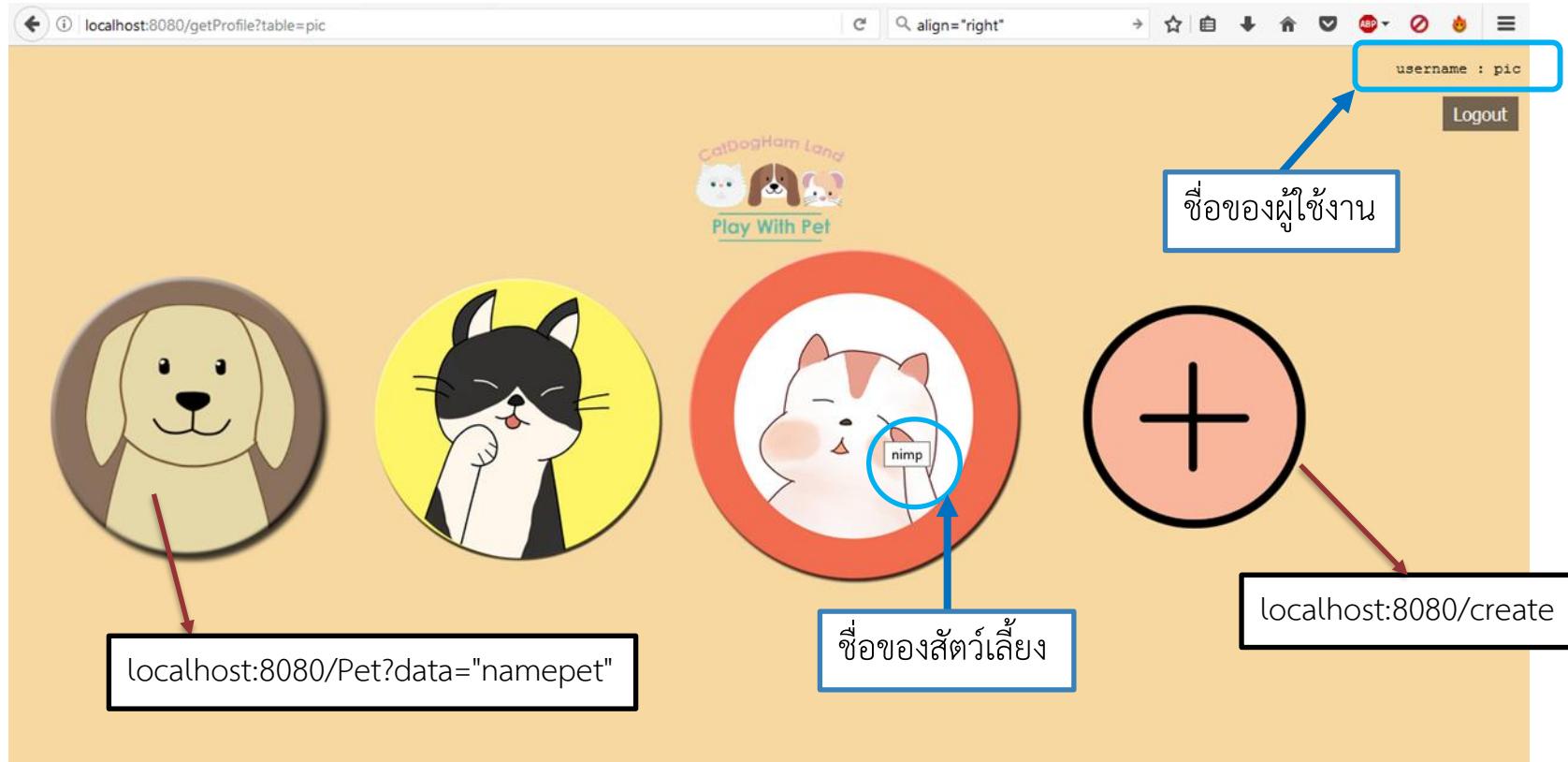
```
app.post('/createPet?', function(req,res){  
    console.log("call regis");  
    res.setHeader('Content-Type', 'application/json');  
    var getjson = getKeys(req.query);  
    //var key = getjson[0] ; //name,pass  
    var value = getjson[1];  
    var dataSplit = value.toString().split(',');  
    var namePet = dataSplit[0];  
    var type = dataSplit[1];  
    //var data = getValuejson(req.query,key);  
    //console.log("key="+name);  
    //console.log("name="+namepath);  
    //console.log("type="+type);  
    //regis(name,pass);  
    createPet(namePet,type);  
    res.send(NamePet);  
    //console.log("json="+json);  
});
```

Table ของ Pet ใน Database MySQL

user	pet_name	type	exp	time	action
pepi	sac	Dog	1	2017-06-03 16:50:08	FEED
pepi	sac	Dog	1	2017-06-03 16:50:55	FEED
pepi	sac	Dog	1	2017-06-03 16:50:59	PLAY
pepi	sac	Dog	1	2017-06-03 16:51:05	FEED
leon	ivy	Cat	1	2017-06-03 17:35:56	create pet

Profile

เมื่อ login หรือ ออกจากหน้า pet และจะเข้าสู่หน้า profile เพื่อดูข้อมูลสัตว์เลี้ยงของตน โดยเมื่อนำมาสู่ไปร่วมบันรูป จะมีชื่อของสัตว์เลี้ยงที่ได้สร้างไว้ขึ้นมา สามารถเลือกเข้าไปดูสัตว์เลี้ยงของคุณได้



Client Profile

function สร้างตารางข้อมูลของสัตว์เลี้ยง

```
function createTable(data){  
  
    console.log("pet!!");  
    console.log(data);  
  
    for (var i =0; i < data.length;i++){  
        for (var key in data[i]){  
            if(key=="pet_name"){  
                pet_name[i] += data[i][key];  
                console.log("name:"+pet_name[i]);  
            }  
            else if(key == "type"){  
                type[i]= data[i][key];  
                console.log(type[i]);  
                //check_type_draw(type);  
            }  
            else if(key == "user"){  
                user = data[i][key];  
                console.log("user:"+user);  
                document.getElementById("username").innerHTML="username : "+user;  
            }  
        }  
    }  
    check_type_draw(type);  
}  
}
```

function ตรวจสอบชนิดของสัตว์เลี้ยง

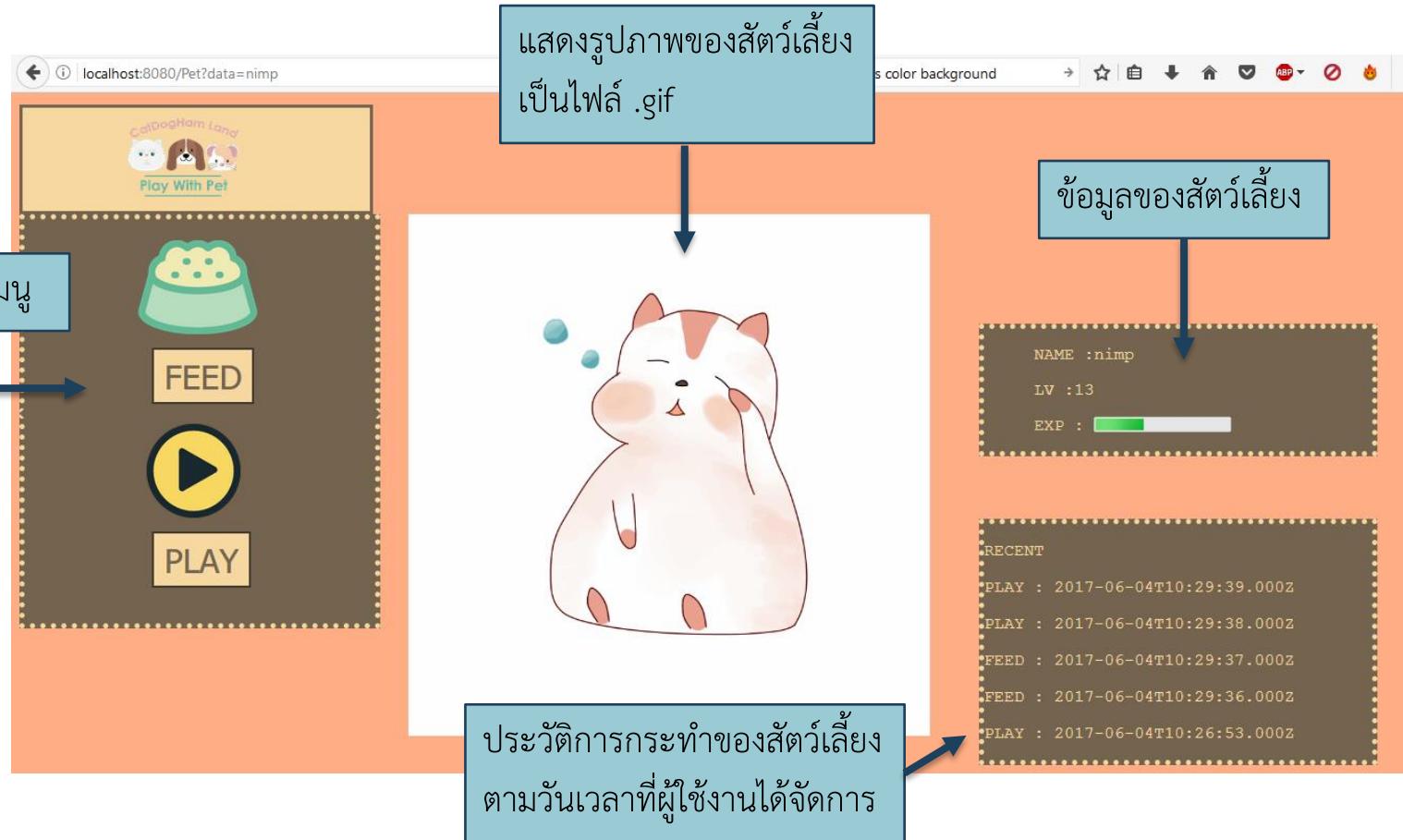
```
function check_type_draw (type){  
    for(var i=0 ; i < type.length;i++){  
        if(type[i]=="Dog"){  
            console.log("d");  
  
            img[i] = document.createElement("IMG");  
            img[i].src = "p-dog.png";  
            img[i].name = pet_name[i];  
            img[i].title = pet_name[i];  
            document.getElementById('imglist').appendChild(img[i]);  
  
            $([ ].concat.apply([], img)).click(function() {  
                namepet = (this.name);  
                text ="http://localhost:8080/Pet?data="+namepet;  
                window.location.href = text ;  
            });  
        }  
        else if(type[i]== "Cat"){  
            console.log("c");  
        }  
    }  
}
```

นำมาแสดงให้ชื่อสัตว์ไม่ซ้ำ

ผู้ Server: Select ข้อมูล

select Distinct
(pet_name),type,user

Pet



- เมื่อเลือกสัตว์เลี้ยงจากหน้า profile จะเข้าสู่หน้า pet ซึ่งจะแสดงข้อมูลต่าง ๆ ของสัตว์เลี้ยง เช่น Level , exp , name, recent
- เมื่อต้องการให้อาหารสัตว์เลี้ยงของคุณให้กดปุ่ม feed จะแสดงรูปสัตว์เลี้ยงของคุณกำลังกินอาหาร และ exp +25
- เมื่อต้องการเล่นกับสัตว์เลี้ยงของคุณให้กดปุ่ม play จะแสดงรูปสัตว์เลี้ยงของคุณพร้อมกับของเล่น และ exp +20

หมายเหตุ เมื่อ exp ครบ 100 Level จะเพิ่มชี้น

Client Pet

```
//call1
ajaxGet('/dataPet',recieveData);
console.log("call");

function gotoProfile(){
    function get_feed(){
//edit new call back
    function get_play(){
        function display_img(){
            function recieveData(data){
                function change_recent(){
                    function change_gif_feed(){
                        function change_gif_play(){
                            function ajaxGet(path,callfunc){
                                function ajaxPost(path, callFunction){
                                    </script>
                                </body>
```

ข้อมูลของ Pet

Filter properties

Object

- action: "PLAY"
- exp: 20
- pet_name: "ivy"
- time: "2017-06-03T19:01:02.000Z"
- type: "Cat"
- user: "pic"

▶ __proto__: Object

Server Pet

server.js

```
/*-----pet-----*/
//SELECT time FROM pet WHERE pet_name='tom' LIMIT 3
app.get('/Pet', function(req, res) {
    var getjson = getKeys(req.query);
    // var key = getjson[0] ; //name,pass
    var value = getjson[1];
    console.log("value="+value);
    //console.log("call page");
    getdata_yourPet(value);
    res.sendFile(path.join(__dirname + '/views/pages/pet.html'));
    console.log("call getdata_yourPet");
    //res.sendFile(dataPet);
});
function getdata_yourPet(namepet){
    var text = 'SELECT * FROM `pet` WHERE pet_name = "'+namepet+'" and user='+namepath;
    console.log(text);
    connection.query(text, function(err, rows) {
        if (err){
            throw err;
        }
        else{
            console.log("getdatapet");
            dataPet=JSON.stringify(rows);
            namepet_pet=namepet;
        }
    });
}
app.get('/dataPet', function(req, res){
```

Server Pet

Server Pet(feed) exp+25

```
};

function insertfeed(user,name,type){
    var text = 'INSERT INTO `pet`(`user`, `pet_name`, `type`, `exp`, `time`, `action`) VALUES ("'+user+'","'+name+'","'+type+'", "'25",CURRENT_TIMESTAMP,"FEED")';
    // var text = 'INSERT INTO `pet`(`user`, `pet_name`, `type`, `exp`, `time`, `action`) VALUES ('+ user+', "'+name+'", "'+type+'", "'1 ",CURRENT_TIMESTAMP,"FEED")';
    console.log(text);
}
```

Server Pet(play) exp+20

```
function insertplay(user,name,type){

    var text = 'INSERT INTO `pet`(`user`, `pet_name`, `type`, `exp`, `time`, `action`) VALUES ("'+user+'","'+name+'","'+type+'", "'20",CURRENT_TIMESTAMP,"PLAY")';
    console.log(text);
}
```



การคณวก

การติดตั้งและการใช้งานโปรแกรม

- การติดตั้ง Node.js และ Express
- การใช้งาน Node.js และ Express

การติดตั้งและการใช้งานโปรแกรม

- การติดตั้ง Node.js และ Express

1. ขั้นตอนการติดตั้ง Node.js

- 1.1 ติดตั้ง Node.js โดยเข้าไปที่เว็บ <https://nodejs.org/en/download/> และ download ตามระบบที่ต้องการ

Downloads

Latest LTS Version: v6.10.0 (includes npm 3.10.10)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.



Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binaries (.tar.gz)

Linux Binaries (x86/x64)

Linux Binaries (ARM)

Source Code

32-bit	64-bit
32-bit	64-bit
64-bit	
64-bit	
32-bit	64-bit
ARMv6	ARMv7
node-v6.10.0.tar.gz	

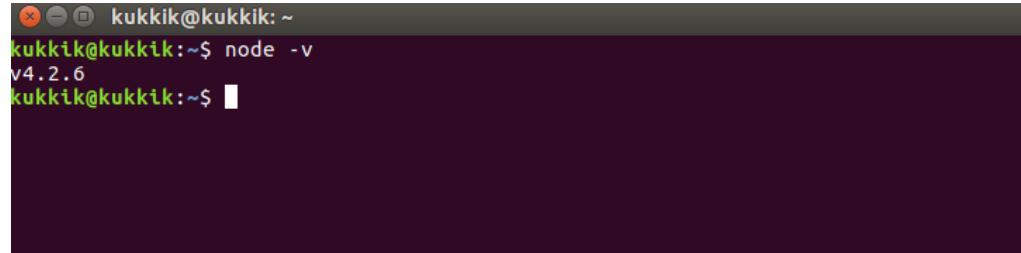
การติดตั้งและการใช้งานโปรแกรม

- การติดตั้ง Node.js และ Express

1.2 เมื่อติดตั้งเสร็จแล้ว เปิด Command-line Interface (Terminal) ขึ้นมา แล้วพิมพ์คำสั่ง

```
node -v
```

จะแสดงเลขเวอร์ชันของ Node.js ขึ้นมา แสดงว่าได้ติดตั้งเสร็จเรียบร้อย



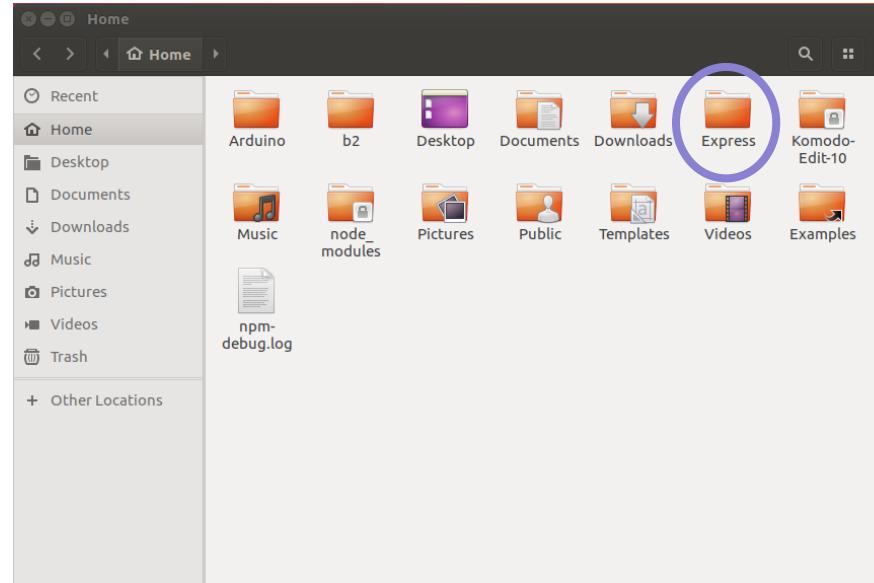
```
kukkik@kukkik:~$ node -v
v4.2.6
kukkik@kukkik:~$
```

การติดตั้งและการใช้งานโปรแกรม

- การติดตั้ง Node.js และ Express

2. ขั้นตอนการติดตั้ง Express

2.1 ติดตั้ง Express โดยเริ่มแรกสร้างไฟล์เดอร์ชื่อ express ขึ้นมา



การติดตั้งและการใช้งานโปรแกรม

• การติดตั้ง Node.js และ Express

2.2 พิมพ์คำสั่งลงใน command ดังนี้

sudo -i

npm init

เพื่อ generate จะได้ไฟล์ package.json มา
ซึ่ง package.json เป็นเสมือนบัตรประจำตัวของ
โปรเจคนั้น ๆ ที่จะระบุว่า โปรเจคนั้นซึ่งว่าอะไร
มีเวอร์ชันที่เท่าไหร่ ใช้ Dependencies อะไรบ้าง
รวมไปถึงรายละเอียดปลีกย่อยอื่น ๆ

```
kukkik@kukkik:~$ sudo -i
[sudo] password for kukkik:
root@kukkik:~# npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.
See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (root)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to /root/package.json:

{
  "name": "root",
  "version": "1.0.0",
  "main": "index.js",
  "dependencies": {
    "body-parser": "^1.17.0",
    "express": "^4.15.0"
  },
  "devDependencies": {
    "express": "^4.15.0"
  },
  "scripts": {
    "test": "echo \\"Warning: no test specified\\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}

Is this ok? (yes) root@kukkik:~#
```

การติดตั้งและการใช้งานโปรแกรม

- การติดตั้ง Node.js และ Express

2.3 เปิด command ขึ้นมา แล้วพิมพ์คำสั่ง

```
npm install express --save
```

ซึ่ง --save คือการบอกว่าตอนจากจะให้ติดตั้ง package นั้นแล้ว ก็ให้ save ชื่อ package นั้นเอาไว้ใน package.json ด้วย เวลาที่จะย้ายไฟล์เดอร์ ก็สามารถติดตั้ง dependencies ห้องหมดได้ ด้วยคำสั่ง npm install

การติดตั้งและการใช้งานโปรแกรม

- การใช้งาน Node.js และ Express

- ในโฟล์เดอร์ express ที่ได้สร้างไว้แล้วก่อนหน้านี้ ซึ่งจะประกอบไปด้วยไฟล์ package.json จากนั้นสร้างไฟล์ server.js

```
Express/
|
|-- server.js
|
|-- node_modules/
|
`-- package.json
```

การติดตั้งและการใช้งานโปรแกรม

- การใช้งาน Node.js และ Express

- เปิด server ด้วยคำสั่ง node server ใน command



```
Command Prompt - node server
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\petch>cd Desktop
C:\Users\petch\Desktop>cd web
C:\Users\petch\Desktop\web>node server
8080 is the magic port
call post
name: kimy
password: 123456
json=
true
namepath="kimy"
table: kimy
value=kimy
call page
call page2
true
data=[{"pet_name":"black","type":"Dog","user":"kimy"}, {"pet_name":"mimi","type":"Cat","user":"kimy"}, {"pet_name":"rosy","type":"Hamster","user":"kimy"}]
yourPet[{"pet_name":"black","type":"Dog","user":"kimy"}, {"pet_name":"mimi","type":"Cat","user":"kimy"}, {"pet_name":"rosy","type":"Hamster","user":"kimy"}]
```

ວິດໂວ

<https://www.youtube.com/watch?v=yFge258qWBE&feature=youtu.be>

