

เอกสารคู่มือระบบ

PHP API

1.) ระบบ Login & Register

1.1) ฟังก์ชันสมัครสมาชิก

```
4 header('Content-Type: application/json');
5 if($_POST['username'] != '' && $_POST['password'] != '' && $_POST['firstname'] != '' && $_POST['lastname'] != '' && $_POST['phone'] != '' && $_POST['role'] != ''){
6     include("dbconnection.php");
7     $conn = dbconnection();
8     // ตรวจสอบการเชื่อมต่อ
9     if ($conn->connect_error) {
10         die("Connection failed: " . $conn->connect_error);
11     }
12     // รับค่าจากการส่งข้อมูลแบบ POST
13     $username = $_POST['username'];
14     $password = $_POST['password'];
15     $firstname = $_POST['firstname'];
16     $lastname = $_POST['lastname'];
17     $phone = $_POST['phone'];
18     $role = $_POST['role'];
19     $hashed_password = password_hash($password, PASSWORD_DEFAULT);
20     $sql = "SELECT * FROM user WHERE username = '$username'";
21     $result = $conn->query($sql);
22     if ($result->num_rows > 0) {
23         $response = array("status" => "already", "message" => "มีบัญชีผู้ใช้นี้อยู่แล้ว");
24         echo json_encode($response);
25     } else {
26         $sql = "INSERT INTO user (username, password, firstname, lastname, phone , role) VALUES ('$username', '$hashed_password', '$firstname', '$lastname', '$phone' , '$role')";
27         if ($conn->query($sql) == TRUE) {
28             // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
29             $response = array("status" => "success", "message" => "สมัครสมาชิกสำเร็จ");
30             echo json_encode($response);
31         } else {
32             // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
33             $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $conn->error);
34             echo json_encode($response);
35         }
36     }
37     // ปิดการเชื่อมต่อ
38     $conn->close();
39 } else {
40     $response = array("status" => "fill_in_blank", "message" => "กรุณากรอกข้อมูลให้ครบ");
41     echo json_encode($response);
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
```

รูปที่ 1.1) register_save.php

ตารางที่ 1.1) Source code การสมัครสมาชิก

บรรทัดที่	รายละเอียด
4 - 51	<p>ตรวจสอบว่ามีข้อมูลที่ส่งมาจากแบบฟอร์มและครบถ้วนหรือไม่ ถ้าข้อมูลใดข้อมูลหนึ่งขาดหาย จะสร้าง JSON response เพื่อแจ้งให้ผู้ใช้กรอกข้อมูลให้ครบ.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ไฟล์ dbconnection.php ซึ่งมีฟังก์ชัน dbconnection() เพื่อเชื่อมต่อ.</p> <p>ตรวจสอบว่าการเชื่อมต่อถูกต้องหรือไม่ ถ้าไม่ถูกต้องจะแสดงข้อความ "Connection failed" และสิ้นสุดการทำงาน.</p> <p>รับค่าที่ส่งมาจากแบบฟอร์มและเก็บไว้ในตัวแปรต่างๆ เช่น username, password, firstname, lastname, phone, และ role.</p>

	<p>ทำการเข้ารหัสผ่านโดยใช้ฟังก์ชัน password_hash() เพื่อเตรียมข้อมูลให้พร้อมสำหรับการเพิ่มลงในฐานข้อมูล.</p> <p>ทำการค้นหา username ในฐานข้อมูล MySQL เพื่อตรวจสอบว่ามี username นี้อยู่แล้วหรือไม่. ถ้าพบ username ที่ซ้ำกันในฐานข้อมูล จะสร้าง JSON response เพื่อแจ้งให้ผู้ใช้ทราบว่า มีบัญชีผู้ใช้นี้อยู่แล้ว.</p> <p>ถ้าไม่พบ username ซ้ำกัน จะทำการเพิ่มข้อมูลผู้ใช้ใหม่ลงในฐานข้อมูล โดยใช้คำสั่ง SQL INSERT INTO.</p> <p>หลังจากที่เพิ่มข้อมูลเสร็จ จะสร้าง JSON response เพื่อแจ้งให้ผู้ใช้ว่าการสมัครสมาชิกสำเร็จ.</p> <p>ถ้ามีปัญหาในการเพิ่มข้อมูลลงในฐานข้อมูล จะสร้าง JSON response เพื่อแจ้งให้ผู้ใช้ทราบว่า มีข้อผิดพลาดเกิดขึ้น.</p>
--	--

1.2) การเข้าสู่ระบบ

```
3
4 if($_POST['username'] != '' && $_POST['password'] != ''){
5     include("dbconnection.php");
6     $conn = dbconnection();
7
8     $username = $_POST['username'];
9     $password = $_POST['password'];
10
11     // เตรียมคำสั่ง SQL เพื่อค้นหา username ในฐานข้อมูล
12     $sql = "SELECT * FROM user WHERE username = '$username'";
13     $result = $conn->query($sql);
14
15     if ($result->num_rows > 0) {
16         // พบ username ในฐานข้อมูล
17         $row = $result->fetch_assoc();
18         // ตรวจสอบว่ารหัสผ่านตรงกันหรือไม่
19         if (password_verify($password, $row['password'])) {
20             // รหัสผ่านตรงกัน
21             // สร้าง JSON response โดยคืนค่า role
22             $response = array("status" => "success", "role" => $row["role"], "id" => $row["id"],
23                             "firstname" => $row["firstname"], "lastname" => $row["lastname"],
24                             "username" => $row["username"], "phone" => $row["phone"]);
25             echo json_encode($response);
26         } else {
27             // รหัสผ่านไม่ตรงกัน
28             // สร้าง JSON response โดยคืนค่า 'ไม่พบบัญชีนี้'
29             $response = array("status" => "no_match_pass", "message" => "รหัสผ่านผิด");
30             echo json_encode($response);
31         }
32     } else {
33         // ไม่พบ username ในฐานข้อมูล
34         // สร้าง JSON response โดยคืนค่า 'ไม่พบบัญชีนี้'
35         $response = array("status" => "no_username", "message" => "ไม่พบบัญชีนี้");
36         echo json_encode($response);
37     }
38
39     // ปิดการเชื่อมต่อ
40     $conn->close();
41 } else {
42     $response = array("status" => "fill_in_blank", "message" => "กรุณากรอกข้อมูลให้ครบ");
43     echo json_encode($response);
44 }
```

รูปที่ 1.2) verify.php

ตารางที่ 1.2) Source code การเข้าสู่ระบบ

บรรทัดที่	รายละเอียด
4 - 45	<p>ส่งข้อมูลผู้ใช้ (ชื่อผู้ใช้และรหัสผ่าน) ผ่านแบบฟอร์มโดยใช้ HTTP POST method.</p> <p>ตรวจสอบว่ามีข้อมูลชื่อผู้ใช้และรหัสผ่านที่ส่งมาหรือไม่ ถ้าไม่มีให้สร้าง JSON response สำหรับแจ้งให้ผู้ใช้กรอกข้อมูลให้ครบ.</p> <p>ถ้ามีข้อมูลที่ส่งมาทั้งชื่อผู้ใช้และรหัสผ่าน ก็ทำการตรวจสอบในฐานข้อมูล ว่ามีชื่อผู้ใช้นี้อยู่หรือไม่.</p> <p>ถ้าพบชื่อผู้ใช้ในฐานข้อมูล จะทำการเปรียบเทียบรหัสผ่านที่ส่งมากับรหัสผ่านที่เก็บในฐานข้อมูล โดยใช้ฟังก์ชัน password_verify.</p> <p>ถ้ารหัสผ่านตรงกันกับที่เก็บในฐานข้อมูล จะสร้าง JSON response ที่ระบุสถานะเป็น "success" พร้อมกับส่งข้อมูลผู้ใช้เพิ่มเติม เช่น บทบาท (role) ของผู้ใช้ และข้อมูลส่วนตัวอื่นๆ.</p>

	<p>ถ้ารหัสผ่านไม่ตรงกัน จะสร้าง JSON response ที่ระบุสถานะเป็น "no_match_pass" และแจ้งให้ผู้ใช้ทราบว่ารหัสผ่านผิด.</p> <p>ถ้าไม่พบชื่อผู้ใช้ในฐานข้อมูล จะสร้าง JSON response ที่ระบุสถานะเป็น "no_username" และแจ้งให้ผู้ใช้ทราบว่าไม่พบบัญชีนี้ในระบบ.</p> <p>หลังจากที่ดำเนินการเสร็จสิ้นทั้งหมด จะปิดการเชื่อมต่อกับฐานข้อมูล MySQL.</p>
--	--

2.) ระบบจัดการบัญชีผู้ใช้

2.1) การดึงข้อมูลผู้ใช้งานทั้งหมดจาก database

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["iddelete"])){
8     $idd = $_GET["iddelete"];
9     $sql = "DELETE FROM `user` WHERE id = $idd";
10    mysqli_query($con, $sql);
11 }
12
13 $query = "SELECT * FROM `user` WHERE 1";
14 $exe = mysqli_query($con,$query);
15
16 $arr = [];
17 while($row=mysqli_fetch_array($exe))
18 {
19     $arr[]=$row;
20 }
21
22 $json = json_encode($arr);
23 // Output the JSON
24 echo $json;
25
26 ?>
```

รูปที่ 2.1) view_user.php

ตารางที่ 2.1) Source code การดึงข้อมูลผู้ใช้งานทั้งหมดจาก database

บรรทัดที่	รายละเอียด
1 - 26	<p>ตรวจสอบว่ามีการส่งค่า "iddelete" มาหรือไม่ หากมี จะเก็บค่า ID ที่ต้องการลบลงในตัวแปร \$idd และทำการลบข้อมูลที่มี ID เท่ากับ \$idd ออกจากตาราง "user" ในฐานข้อมูล.</p> <p>สร้างคำสั่ง SQL สำหรับดึงข้อมูลผู้ใช้งานทั้งหมดจากตาราง "user" ในฐานข้อมูล.</p> <p>ทำการดึงข้อมูลโดยใช้ฟังก์ชัน mysqli_query เพื่อทำการส่งคำสั่ง SQL ไปยัง MySQL และเก็บผลลัพธ์ในตัวแปร \$exe.</p> <p>สร้างตัวแปร \$arr เพื่อเก็บข้อมูลที่ดึงมาจากฐานข้อมูล โดยใช้ลูป while เพื่อวนลูปผลลัพธ์ที่ได้รับจากการดึงข้อมูล.</p> <p>ทำการแปลงข้อมูลที่อ่านมาจากฐานข้อมูลให้เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode และเก็บไว้ในตัวแปร \$json.</p> <p>ส่งข้อมูล JSON ที่ได้มาในขั้นตอนก่อนหน้านี้ออกไปยัง client โดยใช้คำสั่ง echo.</p>

2.2) การดึงข้อมูลผู้ใช้รายคนสำหรับ mobile application

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 $id = $_GET["id"];
8
9 $query = "SELECT * FROM `user` WHERE id = '$id'";
10 $exe = mysqli_query($con,$query);
11
12 $arr = [];
13 while($row=mysqli_fetch_array($exe))
14 {
15     $arr[]=$row;
16 }
17
18 $json = json_encode($arr);
19 // Output the JSON
20 echo $json;
21
22 ?>
```

รูปที่ 2.2) get_user.php

ตารางที่ 2.2) Source code การดึงข้อมูลผู้ใช้รายคนสำหรับ mobile application

บรรทัดที่	รายละเอียด
1 - 22	<p>รับค่า ID ของผู้ใช้จากพารามิเตอร์ \$_GET["id"] ซึ่งเป็น ID ของผู้ใช้ที่ต้องการดึงข้อมูล.</p> <p>สร้างคำสั่ง SQL สำหรับดึงข้อมูลของผู้ใช้ที่มี ID ตามที่รับเข้ามาจากพารามิเตอร์.</p> <p>ทำการดึงข้อมูลโดยใช้ฟังก์ชัน mysqli_query เพื่อทำการส่งคำสั่ง SQL ไปยัง MySQL และเก็บผลลัพธ์ในตัวแปร \$exe.</p> <p>สร้างตัวแปร \$arr เพื่อเก็บข้อมูลที่ดึงมาจากรฐานข้อมูล โดยใช้ลูป while เพื่อวนลูปผลลัพธ์ที่ได้รับจากการดึงข้อมูล.</p> <p>ทำการแปลงข้อมูลที่อ่านมาจากรฐานข้อมูลให้เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode และเก็บไว้ในตัวแปร \$json.</p> <p>ส่งข้อมูล JSON ที่ได้มาออกไปยัง client โดยใช้คำสั่ง echo.</p>

3.) ระบบสมุดเก็บประวัติสัตว์เลี้ยง

3.1) การลงทะเบียนสัตว์เลี้ยง

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if($_POST['idpet'] != '' && $_POST['symptom'] != '' && $_POST['treatment'] != ''){
8     $idpet = $_POST['idpet'];
9     $symptom = $_POST['symptom'];
10    $treatment = $_POST['treatment'];
11
12    $sql = "INSERT INTO `pethistory` (`symptom`, `treatment_history`, `date_of_treatment`, `pet_id`)
13        VALUES ('$symptom','$treatment',NOW(),'$idpet')";
14
15    if ($con->query($sql) === TRUE) {
16        // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
17        $response = array("status" => "success", "message" => "เพิ่มประวัติการรักษาสำเร็จ");
18        echo json_encode($response);
19    }else {
20        // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
21        $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $con->error);
22        echo json_encode($response);
23    }
24
25 }else {
26     // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
27     $response = array("status" => "invalid", "message" => "กรุณากรอกข้อมูลให้ครบ");
28     echo json_encode($response);
29 }
```

รูปที่ 3.1) add_pet.php

ตารางที่ 3.1) Source code การลงทะเบียนสัตว์เลี้ยง

บรรทัดที่	รายละเอียด
1 - 29	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>ตรวจสอบว่าข้อมูลที่รับมาจากแบบฟอร์มไม่ว่างหรือไม่ โดยตรวจสอบความถูกต้องของข้อมูลที่เกี่ยวข้องกับการลงทะเบียนสัตว์เลี้ยง เช่น ชื่อสัตว์เลี้ยง, ชื่อเจ้าของ, สายพันธุ์, วันเกิด, น้ำหนัก เป็นต้น.</p> <p>ถ้าข้อมูลที่รับมาไม่ว่าง จะทำการเก็บข้อมูลลงในตัวแปรต่าง ๆ เช่น \$pet_name, \$ownername, \$species, \$breed, \$birthdate, \$weight โดยใช้ \$_POST.</p> <p>สร้างคำสั่ง SQL สำหรับเพิ่มข้อมูลสัตว์เลี้ยงลงในตาราง "pets" ในฐานข้อมูล MySQL.</p> <p>ทำการเรียกใช้เมธอด query() เพื่อ execute คำสั่ง SQL ที่สร้างขึ้น หากคำสั่ง SQL ทำงานสำเร็จ จะสร้าง JSON response สำหรับการลงทะเบียนสำเร็จ และส่งค่า JSON response นี้กลับไปยัง client.</p>

	<p>หากเกิดข้อผิดพลาดในการ execute คำสั่ง SQL จะสร้าง JSON response สำหรับการลงทะเบียนล้มเหลวพร้อมกับข้อความข้อผิดพลาดที่เกิดขึ้น และส่งค่า JSON response กลับไปยัง client.</p> <p>หากข้อมูลที่ได้รับมาจากแบบฟอร์มไม่ครบถ้วน จะสร้าง JSON response สำหรับการลงทะเบียนล้มเหลวพร้อมกับข้อความที่แสดงว่าต้องกรอกข้อมูลให้ครบถ้วน และส่งค่า JSON response กลับไปยัง client.</p>
--	--

3.2) เพิ่มประวัติการรักษา

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if($_POST['idpet'] != '' && $_POST['symptom'] != '' && $_POST['treatment'] != ''){
8     $idpet = $_POST['idpet'];
9     $symptom = $_POST['symptom'];
10    $treatment = $_POST['treatment'];
11
12    $sql = "INSERT INTO `pethistory`(`symptom`, `treatment_history`, `date_of_treatment`, `pet_id`)
13    VALUES ('$symptom','$treatment',NOW(),'$idpet')";
14
15    if ($con->query($sql) == TRUE) {
16        // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
17        $response = array("status" => "success", "message" => "เพิ่มประวัติการรักษาสำเร็จ");
18        echo json_encode($response);
19    } else {
20        // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
21        $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $con->error);
22        echo json_encode($response);
23    }
24 } else {
25     // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
26     $response = array("status" => "invalid", "message" => "กรุณกรอกข้อมูลให้ครบ");
27     echo json_encode($response);
28 }
29 }
```

รูปที่ 3.2) add_pet_history.php

ตารางที่ 3.2) Source code เพิ่มประวัติการรักษา

บรรทัดที่	รายละเอียด
1 - 28	<p>ตรวจสอบว่าข้อมูลที่รับมาจากแบบฟอร์มไม่ว่างหรือไม่ โดยตรวจสอบความถูกต้องของข้อมูลที่เกี่ยวข้องกับประวัติการรักษาสัตว์เลี้ยง เช่น ID ของสัตว์เลี้ยง, อาการ, ประวัติการรักษา เป็นต้น.</p> <p>ถ้าข้อมูลที่รับมาไม่ว่าง จะทำการเก็บข้อมูลลงในตัวแปรต่าง ๆ เช่น \$idpet, \$symptom, \$treatment โดยใช้ \$_POST.</p> <p>สร้างคำสั่ง SQL สำหรับเพิ่มข้อมูลประวัติการรักษาสัตว์เลี้ยงลงในตาราง "pethistory" ในฐานข้อมูล MySQL โดยเพิ่มวันที่ของการรักษาเข้าไปด้วย.</p> <p>ทำการเรียกใช้เมธอด query() เพื่อ execute คำสั่ง SQL ที่สร้างขึ้น หากคำสั่ง SQL ทำงานสำเร็จ จะสร้าง JSON response สำหรับการเพิ่มประวัติการรักษาสำเร็จ และส่งค่า JSON response นี้กลับไปยัง client.</p> <p>หากเกิดข้อผิดพลาดในการ execute คำสั่ง SQL จะสร้าง JSON response สำหรับการเพิ่มประวัติการรักษาล้มเหลวพร้อมกับข้อความข้อผิดพลาดที่เกิดขึ้น และส่งค่า JSON response กลับไปยัง client.</p> <p>หากข้อมูลที่รับมาจากแบบฟอร์มไม่ครบถ้วน จะสร้าง JSON response สำหรับการเพิ่มประวัติการรักษาล้มเหลวพร้อมกับข้อความที่แสดงว่าต้องกรอกข้อมูลให้ครบถ้วน และส่งค่า JSON response กลับไปยัง client.</p>

3.3) การดูข้อมูลสัตว์เลี้ยงของลูกค้าแต่ละคน

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["owner_name"])){
8     $owner_name = $_GET['owner_name'];
9
10    $sql = "SELECT `pet_id`, `pet_name`, `species`, `breed`, `owner_name`,
11            `weight` FROM `pets` WHERE owner_name = '$owner_name'";
12
13    $exe = mysqli_query($con,$sql);
14
15    $arr = [];
16    while($row=mysqli_fetch_array($exe))
17    {
18        $arr[]=$row;
19    }
20
21    $json = json_encode($arr);
22    // Output the JSON
23    echo $json;
24 }
25 ?>
```

รูปที่ 3.3) view_pet.php

ตารางที่ 3.3) Source code การดูข้อมูลสัตว์เลี้ยงของลูกค้ารายคน

บรรทัดที่	รายละเอียด
1 - 25	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>ตรวจสอบว่ามีค่า owner_name ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["owner_name"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี owner_name ที่ส่งมาผ่าน URI จะทำการดึงข้อมูลสัตว์เลี้ยงของเจ้าของนั้นโดยใช้คำสั่ง SQL จากตาราง "pets" โดยค้นหาด้วย owner_name ที่ระบุ</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>

3.4) ดูข้อมูลสัตว์เลี้ยงรายตัว

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["idpet"])){
8     $id_pet = $_GET['idpet'];
9
10    $sql = "SELECT * FROM `pets` WHERE pet_id = '$id_pet'";
11
12    $exe = mysqli_query($con,$sql);
13
14    $arr = [];
15    while($row=mysqli_fetch_array($exe))
16    {
17        $arr[]=$row;
18    }
19
20    $json = json_encode($arr);
21    // Output the JSON
22    echo $json;
23 }
24 ?>
```

รูปที่ 3.4) view_eachpet_data.php

ตารางที่ 3.4) Source code การดูประวัติการรักษาสัตว์เลี้ยงตาม ID สัตว์เลี้ยง

บรรทัดที่	รายละเอียด
1 - 24	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>รวมไฟล์ "dbconnection.php" เพื่อทำการเชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection().</p> <p>ตรวจสอบว่ามีค่า idpet ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["idpet"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี idpet ที่ส่งมาผ่าน URI จะทำการสร้างคำสั่ง SQL สำหรับดึงข้อมูลของสัตว์เลี้ยงที่มี pet_id ตรงกับที่ระบุ</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>

3.5) การดูประวัติการรักษาสัตว์เลี้ยงตาม ID สัตว์เลี้ยง

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["idpet"])){
8     $id_pet = $_GET['idpet'];
9
10    $sql = "SELECT * FROM `pethistory` WHERE pet_id = '$id_pet' ORDER BY date_of_treatment DESC";
11
12    $exe = mysqli_query($con,$sql);
13
14    $arr = [];
15    while($row=mysqli_fetch_array($exe))
16    {
17        $arr[]=$row;
18    }
19
20    $json = json_encode($arr);
21    // Output the JSON
22    echo $json;
23 }
24 ?>
```

รูปที่ 3.5) view_pet_data.php

ตารางที่ 3.5) Source code การดูประวัติการรักษาสัตว์เลี้ยงตาม ID สัตว์เลี้ยง

บรรทัดที่	รายละเอียด
1 - 24	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>ตรวจสอบว่ามีค่า idpet ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["idpet"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี idpet ที่ส่งมาผ่าน URI จะทำการดึงข้อมูลประวัติการรักษาสัตว์เลี้ยงโดยใช้คำสั่ง SQL จากตาราง "pethistory" โดยค้นหาด้วย pet_id ที่ระบุ โดยจัดเรียงข้อมูลตามวันที่การรักษาล่าสุดลงมาก่อน</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>

4.) ระบบขาย

4.1) เพิ่มสินค้าเข้าตะกร้าชั่วคราว

```
1 <?php
2 header('Content-Type: application/json');
3 |
4 include("dbconnection.php");
5 $conn = dbconnection();
6
7 $product_id = $_POST['product_id'];
8 $product_name = $_POST['product_name'];
9 $product_stock = $_POST['product_stock'];
10 $product_price = $_POST['product_price'];
11 $past_stock = $_POST['past_stock'];
12
13 $sql = "INSERT INTO `livebasket` (`product_id`, `product_name`, `product_stock`, `product_price`, `past_stock`)
14     VALUES ('$product_id', '$product_name', '$product_stock', '$product_price', '$past_stock')";
15
16 if ($conn->query($sql) === TRUE) {
17     // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
18     $response = array("status" => "success", "message" => "เพิ่มลงตะกร้าเรียบร้อยแล้ว");
19     echo json_encode($response);
20 } else {
21     // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
22     $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $conn->error);
23     echo json_encode($response);
24 }
25
```

รูปที่ 4.1) live_basket.php

ตารางที่ 4.1) Source code เพิ่มสินค้าเข้าตะกร้าชั่วคราว

บรรทัดที่	รายละเอียด
1 - 24	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>รับข้อมูลสินค้าที่ผู้ใช้ป้อนผ่านแอปพลิเคชัน ซึ่งประกอบด้วย product_id, product_name, product_stock, product_price, และ past_stock โดยใช้ \$_POST.</p> <p>สร้างคำสั่ง SQL สำหรับเพิ่มข้อมูลสินค้าเข้าตะกร้าสินค้าในตาราง "livebasket" ในฐานข้อมูล MySQL.</p> <p>ทำการ execute คำสั่ง SQL และตรวจสอบว่าเพิ่มข้อมูลสินค้าเข้าตะกร้าสำเร็จหรือไม่.</p> <p>หากเพิ่มข้อมูลสินค้าเข้าตะกร้าสำเร็จ จะสร้าง JSON response สำหรับการเพิ่มสินค้าเข้าตะกร้าสินค้าสำเร็จ และส่งค่า JSON response กลับไปยัง client.</p> <p>หากเกิดข้อผิดพลาดในการ execute คำสั่ง SQL จะสร้าง JSON response สำหรับการเพิ่มสินค้าเข้าตะกร้าสินค้าล้มเหลวพร้อมกับข้อความข้อผิดพลาดที่เกิดขึ้น และส่งค่า JSON response กลับไปยัง client.</p>

4.2) ดูข้อมูลตะกร้าสินค้า

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 $query = "SELECT `product_id`, `product_name`, `product_stock`, `product_price`,
8           `past_stock` FROM `livebasket` ";
9 $exe = mysqli_query($con,$query);
10
11 $arr = [];
12 while($row=mysqli_fetch_array($exe))
13 {
14     $arr[]=$row;
15 }
16
17 $json = json_encode($arr);
18 // Output the JSON
19 echo $json;
20
21 ?>
```

รูปที่ 4.2) view_live_basket.php

ตารางที่ 4.2) Source code ดูข้อมูลตะกร้าสินค้า

บรรทัดที่	รายละเอียด
1 - 21	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>สร้างคำสั่ง SQL สำหรับดึงข้อมูลสินค้าทั้งหมดที่อยู่ในตะกร้าสินค้า โดยค้นหาจากตาราง "livebasket".</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>

4.3) ลบสินค้าในตะกร้าสินค้า

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["id"])){
8     $id = $_GET["id"];
9     $sql = "DELETE FROM `livebasket` WHERE product_id = $id";
10    mysqli_query($con, $sql);
11 }
12 ?>
```

รูปที่ 4.3) delete_live_basket.php

ตารางที่ 4.3) Source code ลบสินค้าในตะกร้าสินค้า

บรรทัดที่	รายละเอียด
1 - 12	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>เชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection() ที่อยู่ในไฟล์ "dbconnection.php".</p> <p>ตรวจสอบว่ามีค่า id ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["id"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี id ที่ส่งมาผ่าน URI จะทำการสร้างคำสั่ง SQL สำหรับลบสินค้าออกจากตะกร้าสินค้าในตาราง "livebasket" โดยค้นหาจาก product_id ที่ระบุ</p> <p>ทำการ execute คำสั่ง SQL และทำการลบสินค้าออกจากตะกร้าสินค้า</p>

4.4) ลบสินค้าใน stock ตามสินค้าในตะกร้า

```
1 <?php
2 header('Content-Type: application/json');
3 include("dbconnection.php");
4 $conn = dbconnection();
5
6 $product_id = $_POST['product_id'];
7 $past_stock = $_POST['past_stock'];
8 $num_of_delete = $_POST['num_of_delete'];
9 $sum = $past_stock - $num_of_delete;
10 $sql = "UPDATE `product` SET `product_stock`='".$sum"' WHERE product_id = $product_id";
11 if ($conn->query($sql) === TRUE) {
12     // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
13     $response = array("status" => "success", "message" => "ทำการสำเร็จ");
14     echo json_encode($response);
15 } else {
16     // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
17     $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $conn->error);
18     echo json_encode($response);
19 }
20
21 $sql = "DELETE FROM livebasket";
22 $conn->query($sql);
23
```

Activate Windows
Go to Settings to activate Windows.

รูปที่ 4.4) update_stock_and_delete_basket.php

ตารางที่ 4.4) Source code ลบสินค้าใน stock ตามสินค้าในตะกร้า

บรรทัดที่	รายละเอียด
1 - 22	<p>รับข้อมูลที่ส่งมาจากแอปพลิเคชัน ซึ่งประกอบด้วย product_id (รหัสสินค้า), past_stock (จำนวนสินค้าที่มีอยู่ก่อนลบ), และ num_of_delete (จำนวนสินค้าที่ต้องการลบ) โดยใช้ \$_POST.</p> <p>คำนวณจำนวนสินค้าในสต็อกหลังการลบโดยลบ num_of_delete จาก past_stock และเก็บผลลัพธ์ไว้ในตัวแปร sum.</p> <p>สร้างคำสั่ง SQL สำหรับการอัปเดตจำนวนสินค้าในตาราง "product" โดยลดจำนวนสินค้าลงตามค่า sum โดยค้นหาจาก product_id.</p> <p>ทำการ execute คำสั่ง SQL และตรวจสอบว่าการอัปเดตสินค้าในสต็อกสำเร็จหรือไม่.</p> <p>หากสำเร็จ จะสร้าง JSON response สำหรับการทำการสำเร็จ และส่งค่า JSON response กลับไปยัง client.</p> <p>หากเกิดข้อผิดพลาดในการ execute คำสั่ง SQL จะสร้าง JSON response สำหรับการล้มเหลว พร้อมกับข้อความข้อผิดพลาด และส่งค่า JSON response กลับไปยัง client.</p> <p>ทำการลบข้อมูลในตาราง "livebasket" เพื่อล้างตะกร้าสินค้าหลังจากทำการอัปเดตสต็อกสำเร็จ โดยใช้คำสั่ง SQL DELETE.</p>

5.) ระบบจัดการคิว

5.1) ระบบจัดการคิว

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6 $year = $_POST['year'];
7 $month = $_POST['month'];
8 $day = $_POST['day'];
9 $id_booker = $_POST['id_booker'];
10 if($_POST['admininsert'] != ''){
11     for ($i = 10; $i <= 16; $i++) {
12         $sql = "INSERT INTO `bookings` (`booking_id`, `booking_datetime`, `queue_status`, `id_booker`)
13             VALUES (NULL, '$year-$month-$day $i:00:00.000000', 'available', '')";
14         mysqli_query($con,$sql);
15     }
16 }
17 if($_POST['booktime'] != ''){
18     $booktime = $_POST['booktime'];
19     if($_POST['cancel'] == 'cancel'){
20         $sql = "UPDATE bookings SET bookings.queue_status='available', bookings.id_booker = ''
21             WHERE bookings.booking_datetime = '$year-$month-$day $booktime'";
22     }else{
23         $sql = "UPDATE bookings SET bookings.queue_status='booked', bookings.id_booker = '$id_booker'
24             WHERE bookings.booking_datetime = '$year-$month-$day $booktime'";
25     }
26     mysqli_query($con,$sql);
27 }
28
29 $sql = "SELECT bookings.booking_datetime, bookings.queue_status, bookings.id_booker FROM bookings WHERE bookings.booking_datetime
30 BETWEEN '$year-$month-$day 10:00:00' AND '$year-$month-$day 18:00:00' ORDER BY bookings.booking_datetime ";
31
32 $exe = mysqli_query($con,$sql);
33
34 $arr = [];
35 while($row=mysqli_fetch_array($exe))
36 {
37     $arr[]=$row;
38 }
39
40 $json = json_encode($arr);
41 // Output the JSON
42 echo $json;
```

รูปที่ 5.1) view_queue.php

ตารางที่ 5.1) Source code ระบบจัดการคิว

บรรทัดที่	รายละเอียด
1 - 42	รับข้อมูลที่ส่งมาจากแอปพลิเคชัน ซึ่งประกอบด้วย year (ปี), month (เดือน), day (วัน), id_booker (รหัสผู้จองคิว) และ admininsert (ค่าสำหรับการเพิ่มข้อมูลจอง) โดยใช้ \$_POST. หากมีการส่งค่า admininsert มา โดยไม่เป็นค่าว่าง จะทำการวนลูปเพื่อสร้างข้อมูลการจองคิวใหม่ สำหรับช่วงเวลาตั้งแต่เวลา 10:00 น. ถึง 16:00 น. แล้วเพิ่มข้อมูลการจองลงในตาราง "bookings" โดยกำหนด queue_status เป็น "available" และ id_booker เป็นค่าว่าง. หากมีการส่งค่า booktime มา โดยไม่เป็นค่าว่าง จะทำการอัปเดตข้อมูลการจองคิวตามเวลาที่กำหนด โดยตรวจสอบว่าเป็นการยกเลิกการจองหรือไม่ หากเป็นการยกเลิก จะทำการอัปเดต queue_status เป็น "available" และ id_booker เป็นค่าว่าง หากไม่ใช่จะอัปเดต queue_status เป็น "booked" และ id_booker เป็นค่า id ของผู้จองคิว.

	<p>ดึงข้อมูลการจองคิวตามวันที่และช่วงเวลาที่กำหนด โดยใช้คำสั่ง SQL SELECT และ ORDER BY เรียงตามเวลาการจองคิว.</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array.</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode.</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo.</p>
--	---

5.2) ดึงข้อมูลนัดหมายของลูกค้า

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["name"])){
8     $name = $_GET['name'];
9
10    $sql = "SELECT * FROM `bookings` WHERE id_booker = '$name' ORDER BY booking_datetime DESC";
11
12    $exe = mysqli_query($con,$sql);
13
14    $arr = [];
15    while($row=mysqli_fetch_array($exe))
16    {
17        $arr[]=$row;
18    }
19
20    $json = json_encode($arr);
21    // Output the JSON
22    echo $json;
23 }
24 ?>
```

รูปที่ 5.2) select_user_appointment.php

ตารางที่ 5.2) Source code ดึงข้อมูลนัดหมายของลูกค้า

บรรทัดที่	รายละเอียด
1 - 24	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>รวมไฟล์ "dbconnection.php" เพื่อทำการเชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection().</p> <p>ตรวจสอบว่ามีค่า name ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["name"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี name ที่ส่งมาผ่าน URI จะทำการสร้างคำสั่ง SQL สำหรับดึงข้อมูลการนัดหมายที่มี id_booker ตรงกับที่ระบุ โดยเรียงลำดับตามเวลาการนัดหมายจากใหม่ไปเก่า</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>

5.) ระบบ stock

5.1) จัดการ stock (ดู stock, ลบ stock, เพิ่ม stock)

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $con = dbconnection();
6
7 if(isset($_GET["iddelete"])){
8     $idd = $_GET["iddelete"];
9     $sql = "DELETE FROM `product` WHERE product_id = $idd";
10    mysqli_query($con, $sql);
11 }
12
13 if(isset($_GET["idsearch"])){
14     $ids = $_GET["idsearch"];
15     $query = "SELECT `product_id`, `product_name`, `product_stock`, `product_price`, `product_detail`
16             | FROM `product` WHERE product_id = $ids";
17 }
18 }else{
19     $query = "SELECT `product_id`, `product_name`, `product_stock`, `product_price`, `product_detail`
20             | FROM `product` ";
21 }
22
23 $exe = mysqli_query($con,$query);
24
25 $arr = [];
26 while($row=mysqli_fetch_array($exe))
27 {
28     $arr[]=$row;
29 }
30
31 $json = json_encode($arr);
32 // Output the JSON
33 echo $json;
34
35 ?>
```

รูปที่ 5.1) view_product.php

ตารางที่ 5.1) Source code จัดการ stock (ดู stock, ลบ stock, เพิ่ม stock)

บรรทัดที่	รายละเอียด
1 - 36	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>รวมไฟล์ "dbconnection.php" เพื่อทำการเชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection().</p> <p>ตรวจสอบว่ามีค่า iddelete หรือ idsearch ที่ส่งมาผ่าน URI หรือไม่ โดยใช้ isset(\$_GET["iddelete"]) และ isset(\$_GET["idsearch"]) เพื่อเช็คว่ามีข้อมูลที่ร้องขอมาหรือไม่</p> <p>หากมี iddelete ที่ส่งมาผ่าน URI จะทำการลบสินค้าที่มี product_id ตรงกับที่ระบุ โดยใช้คำสั่ง SQL DELETE.</p>

	<p>หากมี idsearch ที่ส่งมาผ่าน URI จะทำการสร้างคำสั่ง SQL SELECT เพื่อดึงข้อมูลสินค้าที่มี product_id ตรงกับที่ระบุ และหากไม่มี idsearch ส่งมา จะดึงข้อมูลสินค้าทั้งหมด</p> <p>ทำการ execute คำสั่ง SQL และเก็บผลลัพธ์ลงในตัวแปร \$exe</p> <p>วนลูปผลลัพธ์ที่ได้รับจากการ execute และเก็บข้อมูลลงในตัวแปร \$arr เป็นรูปแบบของ array</p> <p>แปลงข้อมูลที่เก็บใน \$arr เป็นรูปแบบ JSON โดยใช้ฟังก์ชัน json_encode</p> <p>ส่งค่า JSON ที่ได้กลับไปยัง client โดยใช้คำสั่ง echo</p>
--	---

5.2) เพิ่มรายการสินค้าใน stock

```
1 <?php
2 header('Content-Type: application/json');
3
4 if($_POST['product_name'] != '' && $_POST['product_stock'] != '' && $_POST['product_price'] != ''){
5     include("dbconnection.php");
6     $conn = dbconnection();
7
8     $product_name = $_POST['product_name'];
9     $product_stock = $_POST['product_stock'];
10    $product_price = $_POST['product_price'];
11    $product_detail = $_POST['product_detail'];
12
13    $sql = "INSERT INTO `product`(`product_name`, `product_stock`, `product_price`, `product_detail`)
14          VALUES ('$product_name','$product_stock','$product_price','$product_detail')";
15
16    if ($conn->query($sql) === TRUE) {
17        // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
18        $response = array("status" => "success", "message" => "เพิ่มสินค้าสำเร็จ");
19        echo json_encode($response);
20    } else {
21        // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
22        $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกครั้ง Error: " . $sql . "<br>" . $conn->error);
23        echo json_encode($response);
24    }
25
26 }else{
27     $response = array("status" => "fill_in_blank", "message" => "กรุณากรอกข้อมูลให้ครบ");
28     echo json_encode($response);
29 }
```

รูปที่ 5.2) add_product.php

ตารางที่ 5.2) Source code เพิ่มรายการสินค้าใน stock

บรรทัดที่	รายละเอียด
1 - 29	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>ตรวจสอบว่ามีค่า product_name, product_stock, และ product_price ที่ส่งมาผ่านการ POST หรือไม่ โดยใช้คำสั่ง if (\$_POST['product_name'] != '' && \$_POST['product_stock'] != '' && \$_POST['product_price'] != '') เพื่อเช็คว่ามีข้อมูลที่จำเป็นสำหรับการเพิ่มสินค้าหรือไม่ หากมีข้อมูลที่ส่งมาครบถ้วน จะทำการสร้างตัวแปร \$conn ขึ้นมาเพื่อทำการเชื่อมต่อกับฐานข้อมูล MySQL.</p> <p>นำข้อมูลที่ส่งมาผ่าน POST ไปเก็บในตัวแปรต่างๆ เช่น \$product_name, \$product_stock, \$product_price, \$product_detail.</p> <p>สร้างคำสั่ง SQL INSERT เพื่อเพิ่มข้อมูลสินค้าลงในตาราง product โดยใช้ค่าที่รับมาจากผู้ใช้งานในตอนนั้นแล้ว.</p> <p>ทำการ execute คำสั่ง SQL โดยใช้เมธอด query() ของอ็อบเจกต์เชื่อมต่อฐานข้อมูล.</p>

	<p>หากการ execute คำสั่ง SQL สำเร็จ จะสร้าง JSON response โดยกำหนดค่า status เป็น "success" และส่งข้อความ "เพิ่มสินค้าสำเร็จ" กลับไปยัง client.</p> <p>หากการ execute คำสั่ง SQL ล้มเหลว จะสร้าง JSON response โดยกำหนดค่า status เป็น "error" และส่งข้อความ "ล้มเหลวโปรดลองอีกรอบ" พร้อมกับข้อความข้อผิดพลาดที่เกิดขึ้น กลับไปยัง client.</p> <p>หากข้อมูลที่ส่งมาไม่ครบถ้วน จะสร้าง JSON response โดยกำหนดค่า status เป็น "fill_in_blank" และส่งข้อความ "กรุณากรอกข้อมูลให้ครบ" กลับไปยัง client.</p>
--	---

5.3) แก้ไขจำนวนสินค้าใน stock

```
1 <?php
2 header('Content-Type: application/json');
3
4 include("dbconnection.php");
5 $conn = dbconnection();
6
7 $product_id = $_POST['product_id'];
8 $product_stock = $_POST['product_stock'];
9
10 $sql = "UPDATE `product` SET `product_stock` = $product_stock WHERE product_id = $product_id";
11
12 if ($conn->query($sql) === TRUE) {
13     // สร้าง JSON response สำหรับการสมัครสมาชิกสำเร็จ
14     $response = array("status" => "success", "message" => "อัปเดตสำเร็จ");
15     echo json_encode($response);
16 } else {
17     // สร้าง JSON response สำหรับการสมัครสมาชิกล้มเหลว
18     $response = array("status" => "error", "message" => "ล้มเหลวโปรดลองอีกรอบ Error: " . $sql . "<br>" . $conn->error);
19     echo json_encode($response);
20 }
```

รูปที่ 5.3) edit_stock.php

ตารางที่ 5.3) Source code แก้ไขจำนวนสินค้าใน stock

บรรทัดที่	รายละเอียด
1 - 20	<p>กำหนด Header เป็น 'Content-Type: application/json' เพื่อระบุว่าเอกสารที่ส่งกลับจะเป็นประเภท JSON.</p> <p>นำเข้าไฟล์ "dbconnection.php" เพื่อทำการเชื่อมต่อกับฐานข้อมูล MySQL โดยใช้ฟังก์ชัน dbconnection().</p> <p>รับข้อมูลที่ส่งมาผ่านการ POST เก็บไว้ในตัวแปร \$product_id และ \$product_stock ซึ่งเป็นรหัสสินค้าและจำนวนสินค้าที่ต้องการแก้ไข.</p> <p>สร้างคำสั่ง SQL UPDATE เพื่อแก้ไขจำนวนสินค้าในตาราง product โดยกำหนดค่าใหม่ของจำนวนสินค้าให้เท่ากับ \$product_stock สำหรับสินค้าที่มีรหัสเท่ากับ \$product_id.</p> <p>ทำการ execute คำสั่ง SQL โดยใช้เมธอด query() ของอ็อบเจกต์เชื่อมต่อฐานข้อมูล.</p> <p>หากการ execute คำสั่ง SQL สำเร็จ จะสร้าง JSON response โดยกำหนดค่า status เป็น "success" และส่งข้อความ "อัปเดตสำเร็จ" กลับไปยัง client.</p> <p>หากการ execute คำสั่ง SQL ล้มเหลว จะสร้าง JSON response โดยกำหนดค่า status เป็น "error" และส่งข้อความ "ล้มเหลวโปรดลองอีกรอบ" พร้อมกับข้อความข้อผิดพลาดที่เกิดขึ้นกลับไปยัง client.</p>

6.) การเชื่อมต่อ database

5.1) การเชื่อมต่อ database

```
1 <?php
2
3 function dbconnection()
4 {
5     $con=mysqli_connect("localhost","id21981351_admin","Janjamsai2545$","id21981351_vet_database");
6     return $con;
7 }
8
9 ?>
```

รูปที่ 5.1) dbconnection.php

ตารางที่ 5.1) Source code จัดการ stock (ดู stock, ลบ stock, เพิ่ม stock)

บรรทัดที่	รายละเอียด
1 - 9	การเชื่อมต่อ database ที่อยู่บน host: 000webhost.com

Desktop Application

1.) หน้า loginscreen.dart

1.1) ฟังก์ชัน login

```
23 Future login() async {
24   try {
25     String uri = 'https://setest123.000webhostapp.com/php_api/verify.php';
26     var res = await http.post(Uri.parse(uri), body: {
27       "username": usernamecontroller.text,
28       "password": passcontroller.text,
29     });
30
31     var response = jsonDecode(res.body);
32     if (response["status"] == "success") {
33       if (response["role"] == "admin") {
34         Navigator.push(
35           context, MaterialPageRoute(builder: (context) => MainView()));
36       } else if (response["role"] == "doctor") {
37         Navigator.push(context,
38           MaterialPageRoute(builder: (context) => MainViewDoctor()));
39       } else if (response["role"] == "manager") {
40         Navigator.push(context,
41           MaterialPageRoute(builder: (context) => MainViewManager()));
42       } else {
43         _showMyDialog('สมาชิกโปรด login ผ่านโทรศัพท์มือถือ');
44       }
45     } else if (response["status"] == "no_match_pass") {
46       _showMyDialog(response['message']);
47       print(response['message']);
48     } else if (response["status"] == "no_username") {
49       _showMyDialog(response['message']);
50       print(response['message']);
51     } else if (response["status"] == "fill_in_blank") {
52       _showMyDialog(response['message']);
53       print(response['message']);
54     }
55   } catch (e) {
56     print(e);
57   }
58 }
```

รูปที่ 1.1) login

ตารางที่ 1.1) ฟังก์ชัน login

บรรทัดที่	รายละเอียด
23 - 58	<p>กำหนด URI ของเว็บเซิร์ฟเวอร์ที่มีสคริปต์สำหรับการตรวจสอบข้อมูลเข้าสู่ระบบที่ต้องการเข้าถึง.</p> <p>ส่งค่า username และ password ผ่านเมธอด POST ไปยัง URI ที่กำหนด โดยใช้ไลบรารี http ในการเชื่อมต่อและส่งคำขอ.</p> <p>รอการตอบกลับจากเซิร์ฟเวอร์และดึงข้อมูลที่ส่งกลับมาในรูปแบบ JSON.</p> <p>ตรวจสอบสถานะของการเข้าสู่ระบบที่ได้รับจากเซิร์ฟเวอร์ หากสำเร็จ (status เป็น "success") จะเข้าสู่หน้าหลักของแอปพลิเคชันตามบทบาทของผู้ใช้ (admin, doctor, manager) ด้วยการเรียกใช้เมธอด Navigator.push เพื่อเปลี่ยนหน้า.</p> <p>หากไม่สำเร็จ จะแสดงหน้าต่างแจ้งเตือน (Dialog) ด้วยเนื้อหาที่เกี่ยวข้องกับสถานะที่ไม่สำเร็จ.</p> <p>หากเกิดข้อผิดพลาดในการเชื่อมต่อหรือการประมวลผลที่เกิดขึ้น จะถูกจัดการโดยการแสดงข้อความผิดพลาดใน console ของแอปพลิเคชัน.</p>

2.) หน้า registerscreen.dart

2.1) ฟังก์ชันสมัครสมาชิก

```
28 Future insertrecord() async {
29   try {
30     String uri =
31       'https://setest123.000webhostapp.com/php_api/register_save.php';
32     var res = await http.post(Uri.parse(uri), body: {
33       "username": usernamecontroller.text,
34       "password": passcontroller.text,
35       "firstname": namecontroller.text,
36       "lastname": lastnamecontroller.text,
37       "phone": phonecontroller.text,
38       "role": 'member',
39     });
40
41     var response = jsonDecode(res.body);
42     if (response["status"] == "fill_in_blank") {
43       _showMyDialog(response['message']);
44     } else if (response["status"] == "success") {
45       _showMyDialog(response['message']);
46     } else if (response["status"] == "error") {
47       _showMyDialog(response['message']);
48     } else if (response["status"] == "already") {
49       _showMyDialog(response['message']);
50     }
51   } catch (e) {
52     print(e);
53   }
54 }
```

รูปที่ 2.1) insertrecord

ตารางที่ 2.1) ฟังก์ชันสมัครสมาชิก

บรรทัดที่	รายละเอียด
23 - 58	<p>กำหนด URI ของเว็บเซิร์ฟเวอร์ที่มีสคริปต์สำหรับการบันทึกข้อมูลการสมัครสมาชิก.</p> <p>ส่งค่า username, password, firstname, lastname, phone และ role ผ่านเมธอด POST ไปยัง URI ที่กำหนด โดยใช้ไลบรารี http ในการเชื่อมต่อและส่งค่าขอ.</p> <p>รอการตอบกลับจากเซิร์ฟเวอร์และดึงข้อมูลที่ส่งกลับมาในรูปแบบ JSON.</p> <p>ตรวจสอบสถานะของการสมัครสมาชิกที่ได้รับจากเซิร์ฟเวอร์ หากสำเร็จ (status เป็น "success") จะแสดงหน้าต่างแจ้งเตือนด้วยข้อความที่ได้รับ.</p> <p>หากไม่สำเร็จ จะแสดงหน้าต่างแจ้งเตือนด้วยข้อความที่เกี่ยวข้องกับสถานะที่ไม่สำเร็จ.</p> <p>หากเกิดข้อผิดพลาดในการเชื่อมต่อหรือการประมวลผลที่เกิดขึ้น จะถูกจัดการโดยการแสดงข้อความผิดพลาดใน console ของแอปพลิเคชัน.</p>

2.) หน้า stock_screen.dart

2.1) ฟังก์ชันดึงข้อมูล stock ทั้งหมด , ค้นหาสินค้า , ลบสินค้า

```
27 Future<void> getrecord(String ids, String idd) async {
28   String uri = "https://setest123.000webhostapp.com/php_api/view_product.php";
29   if (ids != "") {
30     uri =
31       "https://setest123.000webhostapp.com/php_api/view_product.php?idsearch=$ids";
32   } else if (idd != "") {
33     uri =
34       "https://setest123.000webhostapp.com/php_api/view_product.php?iddelete=$idd";
35   }
36
37   try {
38     var response = await http.get(Uri.parse(uri));
39     setState(() {
40       product = jsonDecode(response.body);
41       isLoading = false; // เมื่อโหลดเสร็จสิ้นกำหนด isLoading เป็น false
42       getnum();
43     });
44   } catch (e) {
45     print(e);
46   }
47 }
48
```

รูปที่ 2.1) getrecord

ตารางที่ 2.1) ฟังก์ชันสมัครสมาชิก

บรรทัดที่	รายละเอียด
27 - 46	<p>กำหนด URI ของ API ที่ใช้ในการดึงข้อมูลสินค้า ซึ่งมีรูปแบบเป็น URL ของเว็บไซต์ที่มีสคริปต์สำหรับการดึงข้อมูลสินค้า.</p> <p>ตรวจสอบว่ามีการส่งค่า ids (idsearch) หรือ idd (iddelete) มาหรือไม่ โดยตรวจสอบค่าที่ถูกส่งเข้ามา หากมีการส่งมาให้กำหนด URI ใหม่ให้ตรงกับการค้นหาหรือการลบข้อมูลสินค้า.</p> <p>ใช้ไลบรารี http เพื่อทำการส่งคำขอ HTTP GET ไปยัง URI ที่กำหนด.</p> <p>เมื่อได้รับข้อมูลคำตอบจากเซิร์ฟเวอร์ จะทำการแปลงข้อมูล JSON ที่ได้รับมาเป็น object และกำหนดให้กับตัวแปร product ที่เก็บข้อมูลสินค้า.</p> <p>ตั้งค่าตัวแปร isLoading เป็น false เพื่อระบุว่าการโหลดข้อมูลสินค้าเสร็จสิ้น.</p> <p>เรียกใช้เมธอด getnum() เพื่อดึงจำนวนสินค้าที่ได้รับมาและทำการแสดงผล.</p> <p>หากเกิดข้อผิดพลาดในการดึงข้อมูลหรือการประมวลผลที่เกิดขึ้น จะถูกจัดการโดยการแสดงข้อความผิดพลาดใน console ของแอปพลิเคชัน.</p>

2.2) ฟังก์ชันเพิ่มสินค้า

```
49 Future<void> addrecord(  
50     String name, String stock, String price, String detail) async {  
51     String uri = "https://setest123.000webhostapp.com/php_api/add_product.php";  
52     try {  
53         var res = await http.post(Uri.parse(uri), body: {  
54             "product_name": name,  
55             "product_stock": stock,  
56             "product_price": price,  
57             "product_detail": detail  
58         });  
59  
60         var response = jsonDecode(res.body);  
61         if (response["status"] == "success") {  
62             _showMyDialog(response['message']);  
63             getnum();  
64         } else if (response["status"] == "error") {  
65             _showMyDialog(response['message']);  
66         } else if (response["status"] == "fill_in_blank") {  
67             _showMyDialog(response['message']);  
68         }  
69     } catch (e) {  
70         print(e);  
71     }  
72     uri = "https://setest123.000webhostapp.com/php_api/view_product.php";  
73     try {  
74         var response = await http.get(Uri.parse(uri));  
75         setState() {  
76             product = jsonDecode(response.body);  
77             getnum();  
78         });  
79     } catch (e) {  
80         print(e);  
81     }  
82 }
```

รูปที่ 2.2) addrecord

ตารางที่ 2.2) ฟังก์ชันเพิ่มสินค้า

บรรทัดที่	รายละเอียด
49 - 80	<p>กำหนดฟังก์ชัน addrecord() เพื่อเพิ่มสินค้าใหม่เข้าไปในระบบ โดยมี parameter คือชื่อสินค้า (name), จำนวนสินค้าในสต็อก (stock), ราคาสินค้า (price), รายละเอียดสินค้า (detail).</p> <p>กำหนด URI ของ API ที่ใช้ในการเพิ่มสินค้า ซึ่งมีรูปแบบเป็น URL ของเว็บไซต์ที่มีสคริปต์สำหรับการเพิ่มสินค้า.</p> <p>ใช้ไลบรารี http เพื่อทำการส่งคำขอ HTTP POST ไปยัง URI ที่กำหนด โดยส่งข้อมูลของสินค้าที่ต้องการเพิ่มเข้าไปด้วย.</p> <p>หลังจากได้รับคำตอบจากเซิร์ฟเวอร์ จะแปลงข้อมูล JSON ที่ได้รับมาเป็น object และตรวจสอบค่า status ที่ได้รับ เพื่อดำเนินการต่อไป:</p> <p>หาก status เป็น "success" แสดงว่าการเพิ่มสินค้าสำเร็จ จะแสดง dialog box แจ้งเตือนถึงความสำเร็จและเรียกใช้เมธอด getnum() เพื่ออัปเดตจำนวนสินค้า.</p>

	<p>หาก status เป็น "error" หรือ "fill_in_blank" แสดงว่ามีข้อผิดพลาดเกิดขึ้น และจะแสดงข้อความแจ้งเตือนผ่าน dialog box.</p> <p>หลังจากการเพิ่มสินค้าสำเร็จแล้ว จะทำการดึงข้อมูลสินค้าทั้งหมดอีกครั้งโดยเรียกใช้ API ที่ใช้ในการดึงข้อมูลสินค้าทั้งหมด โดยมีการเรียกใช้ setState เพื่ออัปเดตข้อมูลสินค้าในหน้าแสดงผล.</p>
--	--

2.3) ฟังก์ชันปรับจำนวนสินค้าใน stock

```

84  Future<void> editstock(String id, String stock) async {
85    String uri = "https://setest123.000webhostapp.com/php_api/edit_stock.php";
86    try {
87      var res = await http.post(Uri.parse(uri), body: {
88        "product_id": id,
89        "product_stock": stock,
90      });
91
92      var response = jsonDecode(res.body);
93      if (response["status"] == "success") {
94        _showMyDialog(response['message']);
95        getnum();
96      } else if (response["status"] == "error") {
97        _showMyDialog(response['message']);
98      }
99    } catch (e) {
100     print(e);
101    }
102    uri = "https://setest123.000webhostapp.com/php_api/view_product.php";
103    try {
104      var response = await http.get(Uri.parse(uri));
105      setState(() {
106        product = jsonDecode(response.body);
107        getnum();
108      });
109    } catch (e) {
110     print(e);
111    }
112  }

```

รูปที่ 2.3) editstock

ตารางที่ 2.3) ฟังก์ชันปรับจำนวนสินค้าใน stock

บรรทัดที่	รายละเอียด
84 - 111	<p>กำหนดฟังก์ชัน editstock() เพื่อปรับจำนวนสินค้าในสต็อก โดยรับพารามิเตอร์เป็น ID ของสินค้า (id) และจำนวนสินค้าใหม่ (stock) ที่ต้องการปรับเปลี่ยน.</p> <p>กำหนด URI ของ API ที่ใช้ในการแก้ไขจำนวนสินค้าในสต็อก โดยมีรูปแบบเป็น URL ของเว็บไซต์ที่มีสคริปต์สำหรับการแก้ไขจำนวนสินค้า.</p>

	<p>ใช้ไลบรารี http เพื่อทำการส่งคำขอ HTTP POST ไปยัง URI ที่กำหนด โดยส่งข้อมูล ID ของสินค้าและจำนวนสินค้าใหม่ที่ต้องการเปลี่ยนแปลงด้วย.</p> <p>หลังจากได้รับคำตอบจากเซิร์ฟเวอร์ จะแปลงข้อมูล JSON ที่ได้รับมาเป็น object และตรวจสอบค่า status ที่ได้รับ เพื่อดำเนินการต่อไป:</p> <p>หาก status เป็น "success" แสดงว่าการแก้ไขจำนวนสินค้าสำเร็จ จะแสดง dialog box แจ้งเตือนถึงความสำเร็จและเรียกใช้เมธอด getnum() เพื่ออัปเดตจำนวนสินค้า.</p> <p>หาก status เป็น "error" แสดงว่ามีข้อผิดพลาดเกิดขึ้น และจะแสดงข้อความแจ้งเตือนผ่าน dialog box.</p> <p>หลังจากการแก้ไขจำนวนสินค้าสำเร็จแล้ว จะทำการดึงข้อมูลสินค้าทั้งหมดอีกครั้งโดยเรียกใช้ API ที่ใช้ในการดึงข้อมูลสินค้าทั้งหมด โดยมีการเรียกใช้ setState เพื่ออัปเดตข้อมูลสินค้าในหน้าแสดงผล.</p>
--	---

3.) หน้า selling_screen.dart

3.1) ฟังก์ชันเพิ่มสินค้าลงตะกร้า

```
53 Future<void> add_basket(String id, String name, int stock, String price,
54   String past_stock) async {
55   int newprice = int.parse(price) * stock;
56   sumprice = sumprice + newprice;
57   String uri = "https://setest123.000webhostapp.com/php_api/live_basket.php";
58   try {
59     var res = await http.post(Uri.parse(uri), body: {
60       "product_id": id,
61       "product_name": name,
62       "product_stock": stock.toString(),
63       "product_price": newprice.toString(),
64       "past_stock": past_stock,
65     });
66
67     var response = jsonDecode(res.body);
68     if (response["status"] == "success") {
69       _showMyDialog(response['message']);
70     } else if (response["status"] == "error") {
71       _showMyDialog(response['message']);
72     }
73   } catch (e) {
74     print(e);
75   }
76   uri = "https://setest123.000webhostapp.com/php_api/view_live_basket.php";
77   try {
78     var response = await http.get(Uri.parse(uri));
79     setState(() {
80       livebasket = jsonDecode(response.body);
81     });
82   } catch (e) {
83     print(e);
84   }
85 }
```

รูปที่ 3.1) add_basket

ตารางที่ 3.1) ฟังก์ชันเพิ่มสินค้าลงตะกร้า

บรรทัดที่	รายละเอียด
53 - 84	<p>กำหนดฟังก์ชัน add_basket() เพื่อเพิ่มสินค้าลงในตะกร้าสินค้า โดยรับพารามิเตอร์เป็น ID สินค้า (id), ชื่อสินค้า (name), จำนวนสินค้า (stock), ราคาต่อหน่วย (price), และจำนวนสินค้าก่อนที่จะเพิ่มลงในตะกร้า (past_stock).</p> <p>คำนวณราคารวมของสินค้าใหม่โดยการคูณราคาต่อหน่วยกับจำนวนสินค้าที่เพิ่มเข้ามาในตะกร้า (price * stock).</p> <p>เพิ่มราคารวมของสินค้าใหม่ลงในตัวแปร sumprice ที่ใช้เก็บราคารวมของสินค้าทั้งหมดในตะกร้า.</p> <p>กำหนด URI ของ API ที่ใช้ในการเพิ่มสินค้าลงในตะกร้าสินค้า โดยมีรูปแบบเป็น URL ของเว็บไซต์ที่มีสคริปต์สำหรับการเพิ่มสินค้าลงในตะกร้า.</p> <p>ใช้ไลบรารี http เพื่อทำการส่งคำขอ HTTP POST ไปยัง URI ที่กำหนด โดยส่งข้อมูล ID สินค้า, ชื่อสินค้า, จำนวนสินค้า, ราคารวม, และจำนวนสินค้าก่อนที่จะเพิ่มลงในตะกร้า ด้วย.</p>

	<p>หลังจากได้รับคำตอบจากเซิร์ฟเวอร์ จะแปลงข้อมูล JSON ที่ได้รับมาเป็น object และตรวจสอบค่า status ที่ได้รับ เพื่อดำเนินการต่อไป:</p> <p>หาก status เป็น "success" แสดงว่าการเพิ่มสินค้าลงในตะกร้าสำเร็จ จะแสดง dialog box แจ้งเตือนถึงความสำเร็จ.</p> <p>หาก status เป็น "error" แสดงว่ามีข้อผิดพลาดเกิดขึ้น และจะแสดงข้อความแจ้งเตือนผ่าน dialog box.</p> <p>หลังจากการเพิ่มสินค้าลงในตะกร้าสำเร็จแล้ว จะทำการดึงข้อมูลสินค้าในตะกร้าสินค้าทั้งหมดอีกครั้งโดยเรียกใช้ API ที่ใช้ในการดึงข้อมูลสินค้าในตะกร้า โดยมีการเรียกใช้ setState เพื่ออัปเดตข้อมูลสินค้าในหน้าแสดงผล.</p>
--	--

3.2) ฟังก์ชันลบสินค้าในตะกร้า

```
86
87 Future<void> delete_basket(String id) async {
88     String uri =
89         "https://setest123.000webhostapp.com/php_api/delete_live_basket.php?id=$id";
90     try {
91         await http.get(Uri.parse(uri));
92     } catch (e) {
93         print(e);
94     }
95     uri = "https://setest123.000webhostapp.com/php_api/view_live_basket.php";
96     try {
97         var response = await http.get(Uri.parse(uri));
98         setState(() {
99             livebasket = jsonDecode(response.body);
100         });
101     } catch (e) {
102         print(e);
103     }
104 }
```

รูปที่ 3.2) delete_basket

ตารางที่ 3.2) ฟังก์ชันลบสินค้าในตะกร้า

บรรทัดที่	รายละเอียด
87 - 102	<p>กำหนดฟังก์ชัน delete_basket() เพื่อลบสินค้าออกจากตะกร้าสินค้า โดยรับพารามิเตอร์เป็น ID ของสินค้าที่ต้องการลบ (id).</p> <p>กำหนด URI ของ API ที่ใช้ในการลบสินค้าออกจากตะกร้า โดยมีรูปแบบ URL ของเว็บไซต์ที่มีสคริปต์สำหรับการลบสินค้าออกจากตะกร้า โดยระบุ ID ของสินค้าที่ต้องการลบ.</p> <p>ใช้เมธอด http.get() เพื่อส่งคำขอ HTTP GET ไปยัง URI ที่กำหนด เพื่อลบสินค้าออกจากตะกร้า. หลังจากการลบสินค้าออกจากตะกร้าสำเร็จแล้ว จะทำการดึงข้อมูลสินค้าในตะกร้าสินค้าทั้งหมดอีกครั้งโดยเรียกใช้ API ที่ใช้ในการดึงข้อมูลสินค้าในตะกร้า โดยมีการเรียกใช้ setState เพื่ออัปเดตข้อมูลสินค้าในหน้าแสดงผล.</p>

3.3) ฟังก์ชันปรับจำนวนสินค้าใน stock

```
106 Future<void> changestock() async {
107   var res;
108   String uri =
109     "https://setest123.000webhostapp.com/php_api/update_stock_and_delete_basket.php";
110   try {
111     for (var index = 0; index < livebasket.length; index++) {
112       res = await http.post(Uri.parse(uri), body: {
113         "product_id": livebasket[index]['product_id'],
114         "past_stock": livebasket[index]['past_stock'],
115         "num_of_delete": livebasket[index]['product_stock'],
116       });
117     }
118     var response = jsonDecode(res.body);
119     if (response["status"] == "success") {
120       _showMyDialog(response['message']);
121     }
122   } catch (e) {
123     print(e);
124   }
125   uri = "https://setest123.000webhostapp.com/php_api/view_live_basket.php";
126   try {
127     var response = await http.get(Uri.parse(uri));
128     setState(() {
129       livebasket = jsonDecode(response.body);
130     });
131   } catch (e) {
132     print(e);
133   }
134   sumprice = 0;
135   getrecord("");
136 }
137
```

รูปที่ 3.3) changestock

ตารางที่ 3.3) ฟังก์ชันปรับจำนวนสินค้าใน stock

บรรทัดที่	รายละเอียด
106 - 136	<p>กำหนด URI ของ API ที่ใช้ในการอัปเดตจำนวนสินค้าใน stock และลบสินค้าที่อยู่ในตะกร้า โดยมีรูปแบบ URL ของเว็บไซต์ที่มีสคริปต์สำหรับการอัปเดตจำนวนสินค้าใน stock และลบสินค้าที่อยู่ในตะกร้า.</p> <p>ใช้ loop for เพื่อวนลูปผ่านสินค้าที่อยู่ในตะกร้าสินค้า (livebasket) โดยส่งคำขอ HTTP POST ไปยัง URI ที่กำหนด เพื่ออัปเดตจำนวนสินค้าใน stock และลบสินค้าที่อยู่ในตะกร้า โดยส่งพารามิเตอร์เป็น ID ของสินค้า (product_id), จำนวนสินค้าที่ถูกลบ (num_of_delete), และจำนวนสินค้าที่มีอยู่ก่อนการลบ (past_stock).</p> <p>หลังจากที่ทำการอัปเดตจำนวนสินค้าใน stock และลบสินค้าที่อยู่ในตะกร้าสำเร็จแล้ว จะทำการดึงข้อมูลสินค้าในตะกร้าสินค้าทั้งหมดอีกครั้งโดยเรียกใช้ API ที่ใช้ในการดึงข้อมูลสินค้าในตะกร้า โดยมีการเรียกใช้ setState เพื่ออัปเดตข้อมูลสินค้าในหน้าแสดงผล.</p> <p>ตั้งค่า sumprice เป็น 0 เพื่อรีเซ็ตราคารวมของสินค้าในตะกร้า.</p> <p>เรียกใช้ฟังก์ชัน getrecord() เพื่ออัปเดตข้อมูลสินค้าทั้งหมดหลังจากการอัปเดตและลบสินค้า.</p>

4.) หน้า report_screen.dart

4.1) ฟังก์ชันดึงข้อมูลสัตว์เลี้ยงของลูกค้ารายคน

```
31 Future<void> getrecord() async {  
32   String nametext = name.text;  
33   String uri =  
34     "https://setest123.000webhostapp.com/php_api/view_pet.php?owner_name=$nametext";  
35   try {  
36     var response = await http.get(Uri.parse(uri));  
37     setState(() {  
38       data = jsonDecode(response.body);  
39     });  
40   } catch (e) {  
41     print(e);  
42   }  
43 }
```

รูปที่ 4.1) getrecord

ตารางที่ 4.1) ฟังก์ชันดึงข้อมูลสัตว์เลี้ยงของลูกค้ารายคน

บรรทัดที่	รายละเอียด
31 - 43	<p>นำค่าข้อมูล name.text ซึ่งเป็นชื่อของเจ้าของสัตว์เลี้ยง, มาเก็บไว้ในตัวแปร nametext.</p> <p>กำหนด URI ของ API ที่ใช้ในการดึงข้อมูลสัตว์เลี้ยงของลูกค้ารายคน โดยระบุชื่อของเจ้าของสัตว์เลี้ยงในพารามิเตอร์ owner_name ที่ต่อท้าย URL.</p> <p>ใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI ที่กำหนด เพื่อดึงข้อมูลจากเซิร์ฟเวอร์.</p> <p>เมื่อได้รับข้อมูลตอบกลับจากเซิร์ฟเวอร์ เรียกใช้ jsonDecode() เพื่อแปลงข้อมูล JSON ที่ได้รับมา ให้กลายเป็น Dart objects.</p> <p>สุดท้ายใช้ setState() เพื่ออัปเดตสถานะของตัวแปร data ที่เก็บข้อมูลสัตว์เลี้ยง ซึ่งทำให้สามารถแสดงผลข้อมูลใหม่ได้บนหน้าจอของแอปพลิเคชัน.</p>

4.2) ฟังก์ชันดึงประวัติการรักษาของสัตว์เลี้ยงตาม ID

```
44
45 Future<void> get_history_record() async {
46   String idpet = id_pet.text;
47   String uri =
48     "https://setest123.000webhostapp.com/php_api/view_pet_data.php?idpet=$idpet";
49   try {
50     var response = await http.get(Uri.parse(uri));
51     setState(() {
52       petdata = jsonDecode(response.body);
53     });
54   } catch (e) {
55     print(e);
56   }
57 }
```

รูปที่ 4.2) get_history_record

ตารางที่ 4.2) ฟังก์ชันดึงประวัติการรักษาของสัตว์เลี้ยงตาม ID

บรรทัดที่	รายละเอียด
45 - 56	<p>ค่า ID ของสัตว์เลี้ยงที่ได้จากการป้อนข้อมูลผู้ใช้จะถูกเก็บไว้ในตัวแปร idpet ด้วยการอ่านข้อมูลจาก TextField ที่มีชื่อว่า id_pet.</p> <p>กำหนด URI ของ API ที่ใช้ในการดึงประวัติการรักษาของสัตว์เลี้ยงโดยระบุ ID ของสัตว์เลี้ยงในพารามิเตอร์ idpet ที่ต่อท้าย URL.</p> <p>ใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI ที่กำหนด เพื่อดึงข้อมูลจากเซิร์ฟเวอร์.</p> <p>เมื่อได้รับข้อมูลตอบกลับจากเซิร์ฟเวอร์ เรียกใช้ jsonDecode() เพื่อแปลงข้อมูล JSON ที่ได้รับมาให้เป็น Dart objects.</p> <p>ใช้ setState() เพื่ออัปเดตสถานะของตัวแปร petdata ที่เก็บข้อมูลประวัติการรักษาของสัตว์เลี้ยงซึ่งทำให้สามารถแสดงผลข้อมูลใหม่ได้บนหน้าจอของแอปพลิเคชัน.</p>

4.3) ฟังก์ชันลงทะเบียนสัตว์เลี้ยง

```
80
81 Future<void> addpet() async {
82   String uri = "https://setest123.000webhostapp.com/php_api/add_pet.php";
83   try {
84     var res = await http.post(Uri.parse(uri), body: {
85       "pet_name": pet_name.text,
86       "ownername": name.text,
87       "species": species.text,
88       "breed": breed.text,
89       "birthdate": birthdate.text,
90       "weight": weight.text,
91     });
92     var response = jsonDecode(res.body);
93     if (response["status"] == "success") {
94       _showMyDialog(response['message']);
95     } else if (response["status"] == "error") {
96       _showMyDialog(response['message']);
97     } else if (response["status"] == "already") {
98       _showMyDialog(response['message']);
99     } else if (response["status"] == "invalid") {
100       _showMyDialog(response['message']);
101     }
102   } catch (e) {
103     print(e);
104   }
105 }
106
```

รูปที่ 4.3) addpet

ตารางที่ 4.3) ฟังก์ชันลงทะเบียนสัตว์เลี้ยง

บรรทัดที่	รายละเอียด
81 - 104	<p>ใช้ http.post() เพื่อส่งคำขอ POST ไปยัง URI ที่กำหนด โดยส่งข้อมูลของสัตว์เลี้ยงใหม่ในรูปแบบของ key-value pairs ซึ่งประกอบด้วยชื่อของสัตว์เลี้ยง (pet_name), ชื่อเจ้าของ (ownername), สปีชีส์ (species), พันธุ์ (breed), วันเกิด (birthdate), และน้ำหนัก (weight).</p> <p>เมื่อได้รับข้อมูลตอบกลับจากเซิร์ฟเวอร์ ใช้ jsonDecode() เพื่อแปลงข้อมูล JSON ที่ได้รับมาให้เป็น Dart objects.</p> <p>ตรวจสอบสถานะของการส่งข้อมูลที่ได้รับมา หากสำเร็จ (status เป็น "success") ให้แสดง dialog แจ้งเตือนว่าลงทะเบียนสัตว์เลี้ยงสำเร็จ หากเกิดข้อผิดพลาดหรือข้อมูลไม่ถูกต้อง ให้แสดงข้อความแจ้งเตือนตามสถานะที่ได้รับจากเซิร์ฟเวอร์.</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการประมวลผล ให้แสดงข้อมูลข้อผิดพลาดที่ได้รับไว้ในคอนโซลของแอปพลิเคชัน.</p>

4.4) ฟังก์ชันเพิ่มประวัติการรักษาของสัตว์เลี้ยง

```
58
59 Future<void> addhistory() async {
60   String uri =
61     "https://setest123.000webhostapp.com/php_api/add_pet_history.php";
62   try {
63     var res = await http.post(Uri.parse(uri), body: {
64       "idpet": id_pet.text,
65       "symptom": symptom.text,
66       "treatment": treatment.text
67     });
68     var response = jsonDecode(res.body);
69     if (response["status"] == "success") {
70       _showMyDialog(response['message']);
71     } else if (response["status"] == "error") {
72       _showMyDialog(response['message']);
73     } else if (response["status"] == "invalid") {
74       _showMyDialog(response['message']);
75     }
76   } catch (e) {
77     print(e);
78   }
79 }
```

รูปที่ 4.4) addhistory

ตารางที่ 4.4) ฟังก์ชันเพิ่มประวัติการรักษาของสัตว์เลี้ยง

บรรทัดที่	รายละเอียด
59 - 77	<p>ใช้ http.post() เพื่อส่งคำขอ POST ไปยัง URI ที่กำหนด โดยส่งข้อมูลของประวัติการรักษาใหม่ในรูปแบบของ key-value pairs ซึ่งประกอบด้วย ID ของสัตว์เลี้ยง (idpet), อาการ (symptom), และการรักษา (treatment).</p> <p>เมื่อได้รับข้อมูลตอบกลับจากเซิร์ฟเวอร์ ใช้ jsonDecode() เพื่อแปลงข้อมูล JSON ที่ได้รับมาให้เป็น Dart objects.</p> <p>ตรวจสอบสถานะของการส่งข้อมูลที่ได้รับมา หากสำเร็จ (status เป็น "success") ให้แสดง dialog แจ้งเตือนว่าเพิ่มประวัติการรักษาสำเร็จ หากเกิดข้อผิดพลาดหรือข้อมูลไม่ถูกต้อง ให้แสดงข้อความแจ้งเตือนตามสถานะที่ได้รับจากเซิร์ฟเวอร์.</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการประมวลผล ให้แสดงข้อมูลข้อผิดพลาดที่ได้รับไว้ในคอนโซลของแอปพลิเคชัน.</p>

5.) หน้า edituser_screen.dart

5.1) การสร้างบัญชีผู้ใช้ของ Manager

```
29
30 Future insertrecord() async {
31   try {
32     String uri =
33       'https://setest123.000webhostapp.com/php_api/register_save.php';
34     var res = await http.post(Uri.parse(uri), body: {
35       "username": usernamecontroller.text,
36       "password": passcontroller.text,
37       "firstname": namecontroller.text,
38       "lastname": lastnamecontroller.text,
39       "phone": phonecontroller.text,
40       "role": role,
41     });
42
43     var response = jsonDecode(res.body);
44     if (response["status"] == "fill_in_blank") {
45       _showMyDialog(response['message']);
46     } else if (response["status"] == "success") {
47       _showMyDialog(response['message']);
48     } else if (response["status"] == "error") {
49       _showMyDialog(response['message']);
50     } else if (response["status"] == "already") {
51       _showMyDialog(response['message']);
52     }
53   } catch (e) {
54     print(e);
55   }
56 }
```

รูปที่ 5.1) insertrecord

ตารางที่ 5.1) Source code การสร้างบัญชีผู้ใช้ของผู้จัดการร้าน

บรรทัดที่	รายละเอียด
30 - 55	เริ่มต้นด้วยการกำหนด URI ของ API ที่จะใช้ในการส่งข้อมูล ในที่นี้คือ 'https://setest123.000webhostapp.com/php_api/register_save.php'. ใช้ http.post เพื่อทำการส่งคำขอ HTTP POST ไปยัง URI ดังกล่าว และส่งข้อมูลด้วย พารามิเตอร์ body ที่เป็น Map ของข้อมูลที่ต้องการส่ง ซึ่งประกอบด้วย: "username": ชื่อผู้ใช้ "password": รหัสผ่าน "firstname": ชื่อ "lastname": นามสกุล "phone": เบอร์โทรศัพท์ "role": บทบาทของผู้ใช้

	<p>เมื่อได้รับการตอบกลับจากเว็บ API ใช้ <code>jsonDecode</code> เพื่อแปลงข้อมูลที่ได้รับเป็นรูปแบบ JSON เป็น Map ของ Dart.</p> <p>ตรวจสอบสถานะของการทำงานที่ได้รับจากเว็บ API โดยใช้ค่าใน key "status" ภายใน Map:</p> <p>หากสถานะเป็น "fill_in_blank" จะแสดงข้อความที่ได้รับจาก key "message" โดยใช้ฟังก์ชัน <code>_showMyDialog</code>.</p> <p>หากสถานะเป็น "success" หรือ "error" จะแสดงข้อความเช่นกัน.</p> <p>หากสถานะเป็น "already" คือมีบัญชีผู้ใช้นี้อยู่แล้ว ก็แสดงข้อความเช่นกัน.</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการรับคำตอบจากเว็บ API จะทำการแสดงข้อผิดพลาดที่เกิดขึ้นใน console โดยใช้ <code>print</code>.</p>
--	--

5.2) การดูบัญชีผู้ใช้ทั้งหมดและการลบบัญชีผู้ใช้ในระบบ

```

57
58 Future<void> getrecord(String idd) async {
59   String uri = "https://setest123.000webhostapp.com/php_api/view_user.php";
60   if (idd != "") {
61     uri =
62       "https://setest123.000webhostapp.com/php_api/view_user.php?iddelete=$idd";
63   }
64   try {
65     var response = await http.get(Uri.parse(uri));
66     setState(() {
67       data = jsonDecode(response.body);
68     });
69   } catch (e) {
70     print(e);
71   }
72 }
73

```

รูปที่ 5.2) getrecord

ตารางที่ 5.2) Source code การดูบัญชีผู้ใช้ทั้งหมดและการลบบัญชีผู้ใช้ในระบบ

บรรทัดที่	รายละเอียด
58 - 72	<p>กำหนด URI เริ่มต้นของเว็บ API ที่จะใช้ในการส่งคำขอ HTTP GET รูปแบบเป็น "https://setest123.000webhostapp.com/php_api/view_user.php".</p> <p>ตรวจสอบว่าตัวแปร <code>idd</code> ที่รับเข้ามาไม่เป็นค่าว่าง ถ้าไม่เป็นค่าว่าง จะเพิ่มพารามิเตอร์ "iddelete" เข้าไปใน URI เพื่อให้เว็บ API ทำการดึงข้อมูลของบัญชีผู้ใช้ที่มี ID ที่ต้องการลบออกไป นั่นคือเพิ่ม <code>"?iddelete=\$idd"</code> เข้าไปท้าย URI.</p> <p>ทำการส่งคำขอ HTTP GET ไปยัง URI ที่กำหนด โดยใช้ <code>http.get</code> และรอการตอบกลับ.</p>

	<p>เมื่อได้รับการตอบกลับจากเว็บ API ใช้ <code>jsonDecode</code> เพื่อแปลงข้อมูลที่ได้รับเป็นรูปแบบ JSON เป็น Map ของ Dart.</p> <p>ทำการอัปเดตสถานะของ State ด้วยข้อมูลที่ได้รับ เพื่อให้ UI ทำการแสดงผลข้อมูลใหม่ที่ได้มา.</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการรับคำตอบจากเว็บ API จะทำการแสดงข้อผิดพลาดที่เกิดขึ้นใน console โดยใช้ <code>print</code>.</p>
--	---

6.) หน้า bookqueuescreen.dart

6.1) ฟังก์ชันการดู, จองคิว, ลบคิว

```
39 Future<void> fetchData() async {
40   String uri = "https://setest123.000webhostapp.com/php_api/view_queue.php";
41   // String uri = "http://127.0.0.1/php_api/view_queue.php";
42   try {
43     setState(() {
44       isLoading = true; // ตั้งค่า isLoading เป็น true เมื่อเริ่มโหลดข้อมูล
45     });
46     var res = await http.post(Uri.parse(uri), body: {
47       "year": _selectedDay.year.toString(),
48       "month": _selectedDay.month.toString(),
49       "day": _selectedDay.day.toString(),
50       "booktime": booktime,
51       "cancel": cancel,
52       "id_booker": name_booker.text,
53       "admininsert": admininsert,
54     });
55     setState(() {
56       data = jsonDecode(res.body);
57       isLoading =
58         false; // ตั้งค่า isLoading เป็น false เมื่อโหลดข้อมูลเสร็จสิ้น
59     });
60   } catch (e) {
61     print(e);
62     setState(() {
63       isLoading =
64         false; // ตั้งค่า isLoading เป็น false ในกรณีที่เกิดข้อผิดพลาด
65     });
66   }
67 }
```

รูปที่ 6.1) fetchData

ตารางที่ 6.1) ฟังก์ชันการดู, จองคิว, ลบคิว

บรรทัดที่	รายละเอียด
39 - 66	<p>กำหนดค่า isLoading เป็น true เพื่อแสดงว่ากำลังโหลดข้อมูล.</p> <p>ส่งคำขอ POST ไปยัง URI ที่กำหนด โดยส่งข้อมูลที่เกี่ยวข้องกับวันที่ที่เลือก (_selectedDay) และการจองคิวที่ผู้ใช้เลือก (booktime, cancel (กรณียกเลิก), id_booker)</p> <p>เมื่อได้รับข้อมูลตอบกลับจากเซิร์ฟเวอร์ แปลงข้อมูล JSON ที่ได้รับมาให้กลายเป็น Dart objects และเก็บไว้ในตัวแปร data.</p> <p>ตั้งค่า isLoading เป็น false เพื่อแสดงว่าการโหลดข้อมูลเสร็จสิ้น</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการประมวลผล ตั้งค่า isLoading เป็น false เพื่อแสดงว่าการโหลดข้อมูลเสร็จสิ้นและแสดงข้อผิดพลาดในการดึงข้อมูลในคอนโซล</p>

Mobile Application

1.) หน้า loginscreen.dart

1.1) ฟังก์ชัน login

```
23
24 Future login() async {
25   try {
26     String uri = 'https://setest123.000webhostapp.com/php_api/verify.php';
27     var res = await http.post(Uri.parse(uri), body: {
28       "username": usernamecontroller.text,
29       "password": passcontroller.text,
30     });
31
32     var response = jsonDecode(res.body);
33     if (response["status"] == "success") {
34       if (response["role"] == "member") {
35         Navigator.push(
36           context, MaterialPageRoute(builder: (context) => RountScreen());
37         saveUserData(response["id"], response["firstname"],
38           response["lastname"], response["username"], response["phone"]);
39       } else {
40         _showMyDialog("รหัสผ่านไม่ถูกต้อง");
41       }
42     } else if (response["status"] == "no_match_pass") {
43       _showMyDialog(response['message']);
44       print(response['message']);
45     } else if (response["status"] == "no_username") {
46       _showMyDialog(response['message']);
47       print(response['message']);
48     } else if (response["status"] == "fill_in_blank") {
49       _showMyDialog(response['message']);
50       print(response['message']);
51     }
52   } catch (e) {
53     print(e);
54   }
55 }
```

รูปที่ 1.1)

ตารางที่ 1.1)

บรรทัดที่	รายละเอียด
24 - 53	<p>ส่งคำขอ POST ไปยัง URI ที่กำหนด โดยส่งข้อมูลเข้าระบบของผู้ใช้ (ชื่อผู้ใช้และรหัสผ่าน) และรอการตอบกลับ. เมื่อได้รับคำตอบจากเซิร์ฟเวอร์ แปลงข้อมูล JSON ที่ได้รับมาให้กลายเป็น Dart objects และเก็บไว้ในตัวแปร response.</p> <p>ตรวจสอบค่าของ status ในการตรวจสอบว่าการเข้าสู่ระบบเสร็จสมบูรณ์หรือไม่:</p> <p>ถ้า status เป็น "success" และบทบาทของผู้ใช้เป็น "member" ให้เปลี่ยนหน้าไปยังหน้าหลักของแอปพลิเคชันและบันทึกข้อมูลผู้ใช้งานในหน่วยความจำโดยใช้ฟังก์ชัน saveUserData()</p> <p>ถ้า status เป็น "no_match_pass" ให้แสดงกล่องข้อความแจ้งเตือนว่า "รหัสผ่านไม่ถูกต้อง"</p> <p>ถ้า status เป็น "no_username" ให้แสดงกล่องข้อความแจ้งเตือนว่า "ไม่พบชื่อผู้ใช้"</p> <p>ถ้า status เป็น "fill_in_blank" ให้แสดงกล่องข้อความแจ้งเตือนว่า "กรุณากรอกข้อมูลให้ครบ"</p> <p>หากเกิดข้อผิดพลาดในการส่งคำขอหรือการประมวลผล ให้แสดงข้อผิดพลาดในคอนโซล.</p>

1.2) ฟังก์ชันเก็บข้อมูลผู้ใช้งาน hardware

```
57 void saveUserData(String id, String firstname, String lastname,  
58 String username, String phone) async {  
59     SharedPreferences.getInstance().then(  
60         (prefs) {  
61             prefs.setString('id', id);  
62             prefs.setString('firstname', firstname);  
63             prefs.setString('lastname', lastname);  
64             prefs.setString('username', username);  
65             prefs.setString('phone', phone);  
66         },  
67     );  
68 }  
69
```

รูปที่ 1.2) saveUserData

ตารางที่ 1.2) ฟังก์ชันเก็บข้อมูลผู้ใช้งานเครื่องโทรศัพท์

บรรทัดที่	รายละเอียด
57 - 67	การเรียกใช้ SharedPreferences.getInstance() จะสร้างอ็อบเจกต์ของ SharedPreferences ที่ใช้ในการเก็บข้อมูลลงใน hardware จากนั้นเก็บข้อมูลผู้ใช้ที่ดีมาจาก api คือ id, firstname, lastname, username, phone

2.) หน้า menuseen.dart

2.1) ฟังก์ชันดึงข้อมูลนัดหมายของผู้ใช้

```
23 Future<void> getRecord() async {
24   SharedPreferences prefs = await SharedPreferences.getInstance();
25   String? name = prefs.getString('firstname');
26   String uri =
27     "https://setest123.000webhostapp.com/php_api/select_user_appointment.php?name=$name";
28   try {
29     var response = await http.get(Uri.parse(uri));
30     setState(() {
31       appointment = jsonDecode(response.body);
32     });
33   } catch (e) {
34     print(e);
35   }
36   //print(appointment[0]['booking_datetime']);
37 }
38
```

รูปที่ 2.1) getRecord

ตารางที่ 2.1) ดึงข้อมูลนัดหมายของผู้ใช้

บรรทัดที่	รายละเอียด
23 - 36	<p>SharedPreferences.getInstance() ใช้เพื่อให้สามารถเข้าถึง SharedPreferences ได้โดยเรียกใช้ await เพื่อรอให้คืนค่าก่อนที่จะดำเนินการต่อ.</p> <p>จากนั้นเราดึงข้อมูลชื่อของผู้ใช้ (firstname) ออกมาจาก SharedPreferences โดยใช้ prefs.getString('firstname'). ข้อมูลชื่อนี้จะถูกใช้ในการสร้าง URI เพื่อเรียกข้อมูลนัดหมายของผู้ใช้จากเซิร์ฟเวอร์.</p> <p>ต่อมาเราสร้าง URI โดยรวมชื่อของผู้ใช้เข้าไปเพื่อส่งคำขอไปยังเซิร์ฟเวอร์ที่มีเครื่องหมายคำถามเพื่อแทนที่ชื่อผู้ใช้.</p> <p>เมื่อได้ URI แล้ว เราใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI ดังกล่าวเพื่อขอข้อมูลนัดหมายของผู้ใช้จากเซิร์ฟเวอร์.</p> <p>เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ ข้อมูลที่ได้จะถูกแปลงเป็น JSON ด้วย jsonDecode(response.body) และจะถูกเก็บไว้ในตัวแปร appointment.</p> <p>ในท้ายที่สุด เราอัปเดตสถานะของ State ด้วยข้อมูลที่ได้จากการเรียก API เพื่อให้แสดงผลข้อมูลนัดหมายใหม่บนหน้าจอ.</p>

3.) หน้า pet_screen.dart

3.1) ฟังก์ชันดึงข้อมูลสัตว์เลี้ยงที่เป็นเจ้าของ

```
21
22 Future<void> getRecord() async {
23   SharedPreferences prefs = await SharedPreferences.getInstance();
24   String? name = prefs.getString('firstname');
25   String uri =
26     "https://setest123.000webhostapp.com/php_api/view_pet.php?owner_name=$name";
27   try {
28     var response = await http.get(Uri.parse(uri));
29     setState(() {
30       pet = jsonDecode(response.body);
31     });
32   } catch (e) {
33     print(e);
34   }
35 }
36
```

รูปที่ 3.1) getRecord

ตารางที่ 3.1) ฟังก์ชันดึงข้อมูลสัตว์เลี้ยงที่เป็นเจ้าของ

บรรทัดที่	รายละเอียด
22 - 34	<p>SharedPreferences.getInstance() ใช้เพื่อให้สามารถเข้าถึง SharedPreferences ได้โดยเรียกใช้ await เพื่อรอให้คันทิ้งค่าก่อนที่จะดำเนินการต่อ.</p> <p>จากนั้นเราดึงข้อมูลชื่อของผู้ใช้ (firstname) ออกมาจาก SharedPreferences โดยใช้ prefs.getString('firstname'). ข้อมูลชื่อนี้จะถูกใช้ในการสร้าง URI เพื่อเรียกข้อมูลสัตว์เลี้ยงของผู้ใช้จากเซิร์ฟเวอร์.</p> <p>เมื่อได้ชื่อของเจ้าของสัตว์เลี้ยงแล้ว เราสร้าง URI โดยรวมชื่อของเจ้าของสัตว์เลี้ยงเข้าไปเพื่อส่งคำขอไปยังเซิร์ฟเวอร์ที่มีเครื่องหมายคำถามเพื่อแทนที่ชื่อของเจ้าของสัตว์เลี้ยง.</p> <p>เมื่อได้ URI แล้ว เราใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI ดังกล่าวเพื่อขอข้อมูลสัตว์เลี้ยงของผู้ใช้จากเซิร์ฟเวอร์.</p> <p>เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ ข้อมูลที่ได้จะถูกแปลงเป็น JSON ด้วย jsonDecode(response.body) และจะถูกเก็บไว้ในตัวแปร pet.</p> <p>ในท้ายที่สุด เราอัปเดตสถานะของ State ด้วยข้อมูลที่ได้จากการเรียก API เพื่อให้แสดงผลข้อมูลสัตว์เลี้ยงใหม่บนหน้าจอ.</p>

4.) หน้า pet_screen.dart

4.1) ฟังก์ชันดึงข้อมูลรายละเอียดสัตว์เลี้ยง

```
24 Future<void> getRecord() async {  
25   print(widget.id);  
26   String uri =  
27     "https://setest123.000webhostapp.com/php_api/view_eachpet_data.php?idpet=${widget.id}";  
28   try {  
29     var response = await http.get(Uri.parse(uri));  
30     setState(() {  
31       pet = jsonDecode(response.body);  
32     });  
33   } catch (e) {  
34     print(e);  
35   }  
36 }
```

รูปที่ 4.1) getRecord

ตารางที่ 4.1) ฟังก์ชันดึงข้อมูลสัตว์เลี้ยง

บรรทัดที่	รายละเอียด
24 - 35	<p>ในบรรทัดแรกของฟังก์ชัน เราทำการพิมพ์ค่า id ที่ได้รับจาก widget โดยใช้ print(widget.id) เพื่อตรวจสอบว่าเราได้รับ id ให้ถูกต้องหรือไม่ ซึ่งการพิมพ์ค่านี้จะช่วยให้เราตรวจสอบค่า id ในกรณีที่เรามีการเรียกใช้ฟังก์ชันนี้ใน widget หรือหน้าจอที่ต่างกัน</p> <p>จากนั้นเราสร้าง URI โดยรวม id ของสัตว์เลี้ยงลงไปใน URI เพื่อสร้างคำขอที่เรียกข้อมูลสัตว์เลี้ยงจากเซิร์ฟเวอร์ โดยการเข้าถึง id ของสัตว์เลี้ยงจาก widget ในรูปแบบ \${widget.id}</p> <p>เมื่อได้ URI แล้ว เราใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI เพื่อขอข้อมูลสัตว์เลี้ยงตาม id ที่ระบุ</p> <p>เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ ข้อมูลที่ได้จะถูกแปลงเป็น JSON ด้วย jsonDecode(response.body) และจะถูกเก็บไว้ในตัวแปร pet.</p> <p>ในท้ายที่สุด เราอัปเดตสถานะของ State ด้วยข้อมูลที่ได้จากการเรียก API เพื่อให้แสดงผลข้อมูลสัตว์เลี้ยงใหม่บนหน้าจอ.</p>

5.) หน้า pet_history.dart

5.1) ดึงข้อมูลประวัติการรักษาสัตว์เลี้ยง

```
19 Future<void> getRecord() async {
20   String uri =
21     "https://setest123.000webhostapp.com/php_api/view_pet_data.php?idpet=${widget.id}";
22   try {
23     var response = await http.get(Uri.parse(uri));
24     setState(() {
25       data = jsonDecode(response.body);
26     });
27   } catch (e) {
28     print(e);
29   }
30 }
```

รูปที่ 5.1) getRecord

ตารางที่ 5.1) ดึงข้อมูลประวัติการรักษาสัตว์เลี้ยง

บรรทัดที่	รายละเอียด
19 - 29	<p>สร้าง URI โดยรวม id ของสัตว์เลี้ยงลงไปใน URI เพื่อสร้างคำขอที่เรียกข้อมูลการรักษาของสัตว์เลี้ยงจากเซิร์ฟเวอร์ โดยใช้ \${widget.id} เพื่อเข้าถึง id ของสัตว์เลี้ยงที่ได้รับมาจาก widget</p> <p>ใช้ http.get() เพื่อส่งคำขอ GET ไปยัง URI เพื่อขอข้อมูลการรักษาของสัตว์เลี้ยงตาม id ที่ระบุ เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ ข้อมูลที่ได้จะถูกแปลงเป็น JSON ด้วย jsonDecode(response.body) และจะถูกเก็บไว้ในตัวแปร data</p> <p>ในท้ายที่สุด เราอัปเดตสถานะของ State ด้วยข้อมูลที่ได้จากการเรียก API เพื่อให้แสดงผลข้อมูลการรักษาของสัตว์เลี้ยงบนหน้าจอ.</p>