



รายงาน

เรื่อง โครงการระยะที่ 2 การวิเคราะห์และออกแบบเว็บ

เสนอ

ผศ. ดร. จิตาภา ไกรสังข์
อาจารย์ ดร. วุฒิชิต แสงผล

จัดทำโดย

นางสาว	ธัญญารัตน์	วุฒิรุ่งเรืองสกุล	6687063
นางสาว	เขมิสรา	อนรรฆมงคล	6687074
นาย	ชวัลวิทย์	ควรรพพัฒน์กุล	6687078
นาย	นายสิริธีร์	กลิ่นบัว	6687102

คณะเทคโนโลยีสารสนเทศและการสื่อสาร นักศึกษาชั้นปีที่ 2

โครงการนี้เป็นส่วนหนึ่งของรายวิชา

ITDS242 Web Technologies Lab

ภาคการศึกษาที่ 1 ปีการศึกษา 2567

สารบัญ

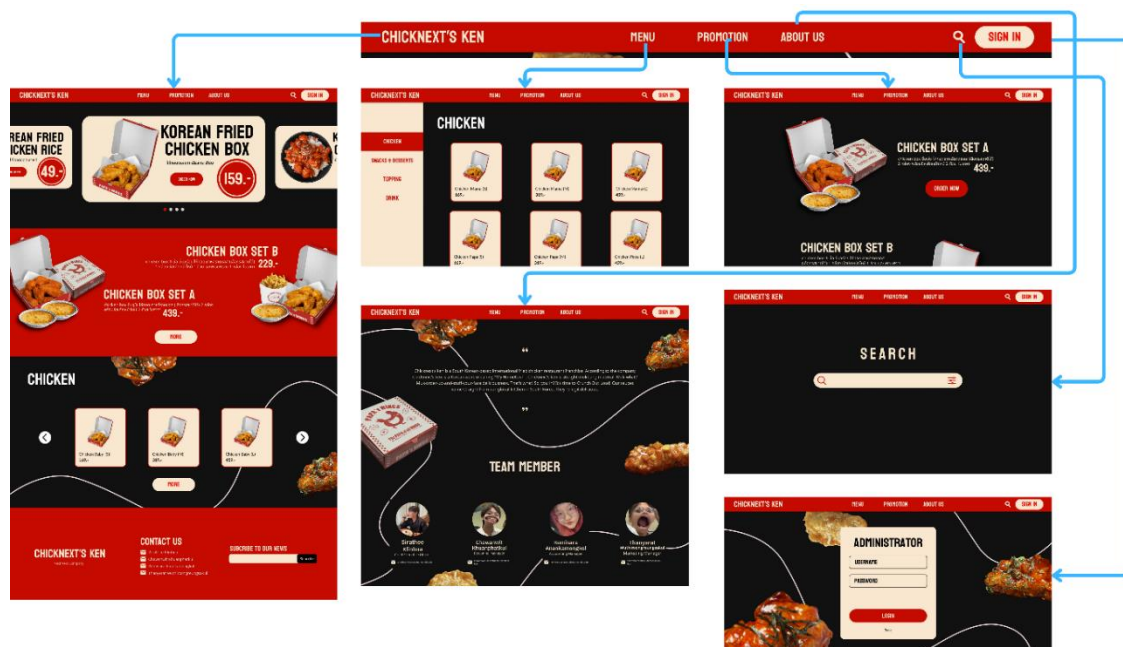
หัวข้อ	หน้า
ข้อมูลองค์รวมของโครงการ	1
แผนผังหน้าต่างๆ ของเว็บแอปพลิเคชัน	1
- แถบนำทางของเว็บไซต์	1
- ส่วนท้ายเว็บไซต์	1
- Home Page	2
- “Login” Page	2
- “Search” page	3
- “Detail” page	4
- “Product/Service Management” page	4
- “User Account Management” page	5
- “Team” page	5
รายละเอียดหน้าเว็บต่างๆ	6
- หน้าหลักของเว็บ (Home Page)	6
- หน้าเว็บสำหรับการเข้าสู่ระบบ (Log-in) ของผู้ดูแลระบบ	7
- หน้าสำหรับการค้นหาข้อมูลเว็บในส่วนสินค้าหรือเซอร์วิส	7
- หน้าสำหรับการแสดงรายละเอียดสินค้าหรือเซอร์วิส	8
- หน้าสำหรับการจัดการข้อมูลของสินค้าหรือเซอร์วิส	8
- หน้าสำหรับการจัดการข้อมูลผู้ดูแลระบบ	10
- หน้า About Us แสดงรายละเอียดสมาชิกในกลุ่ม	12
รายละเอียดของเว็บเซอร์วิส และโค้ด	13
ผลการทดสอบของเว็บเซอร์วิสทั้งหมด	36
แหล่งอ้างอิง	46

ข้อมูลโดยรวมของโครงการ

ธุรกิจร้านอาหาร ChickNext's Ken ถูกก่อตั้งขึ้นจากความตั้งใจของผู้ก่อตั้งที่ต้องการให้ลูกค้าได้ลิ้มลองรสชาติแสนอร่อยของไก่ทอดเกาหลี โดยมีกลุ่มเป้าหมายคือกลุ่มวัยรุ่นที่ชื่นชอบรสชาติอาหารเกาหลีเป็นทุนเดิม เป้าหมายหลักของแพลตฟอร์มนี้จัดทำขึ้นเพื่อให้ลูกค้าสามารถเข้าถึงการสั่งซื้อไก่ทอดเกาหลีได้โดยสะดวก โดยมีขอบเขตการใช้งานในส่วนของลูกค้าคือ สามารถดู ค้นหา สั่งซื้อรายการอาหารได้ และขอบเขตการใช้งานของพนักงานคือ สามารถเพิ่ม ลด แก้ไขรายการอาหารและข้อมูลต่างๆได้ โดยทางคณะผู้จัดทำได้อ้างอิงการออกแบบข้อมูลบางส่วนมาจากเว็บไซต์ร้านอาหาร KFC และ Bonchon เพื่อให้การออกแบบเว็บไซต์มีความสมบูรณ์มากยิ่งขึ้น

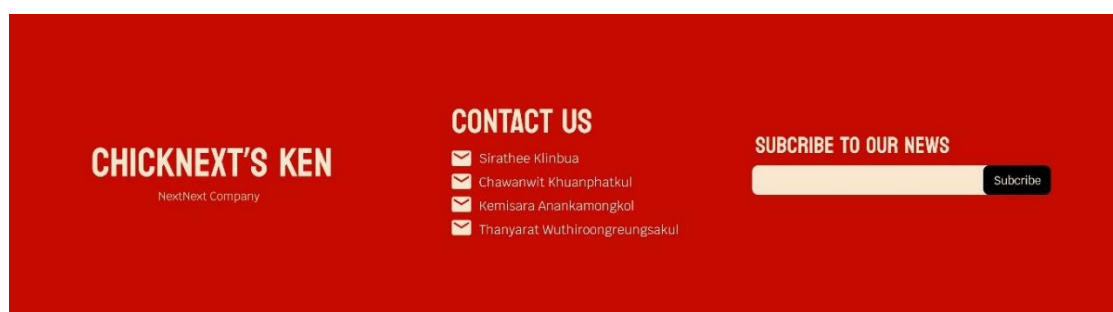
แผนผังหน้าต่างๆ ของเว็บแอปพลิเคชัน

แถบนำทางของเว็บไซต์ (Navigation Bar)



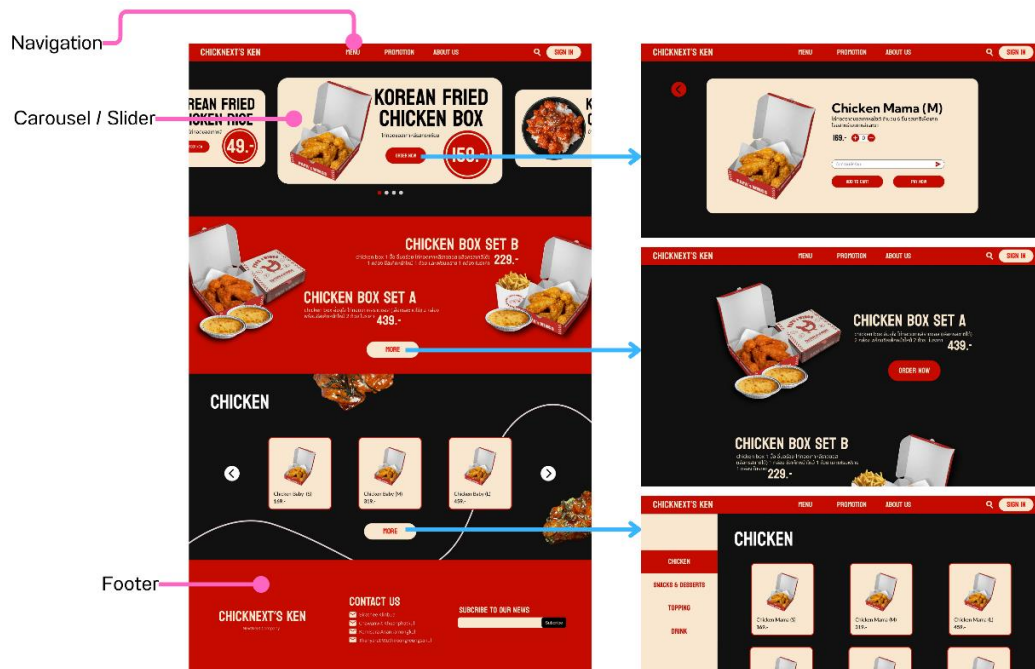
(ภาพที่ 1 แสดงแถบนำทางของเว็บไซต์)

ส่วนท้ายของเว็บไซต์ (Footer)



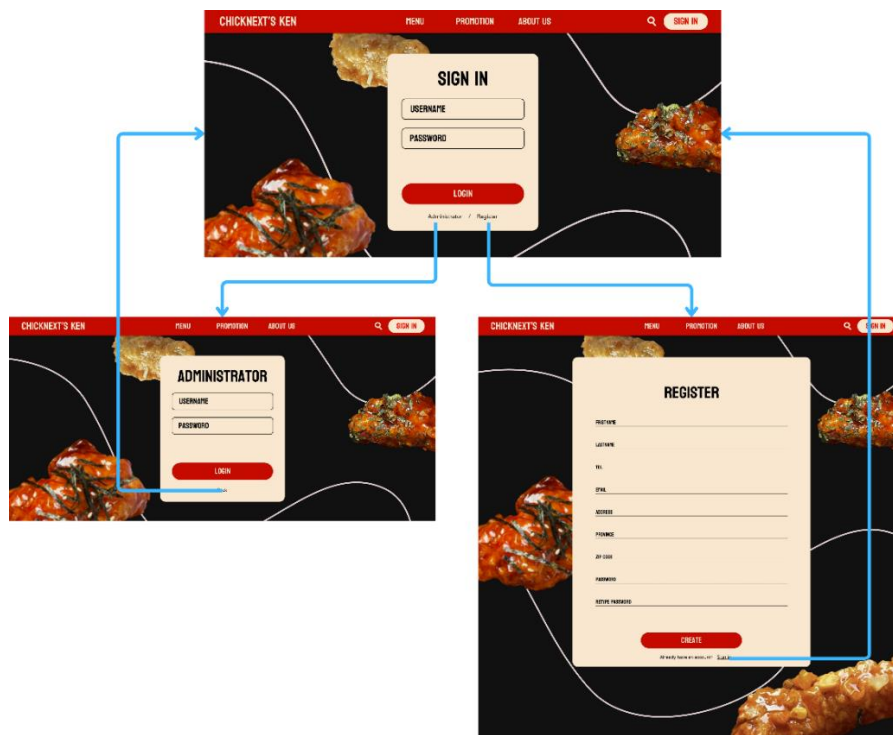
(ภาพที่ 2 แสดงส่วนท้ายเว็บไซต์)

1. Home Page

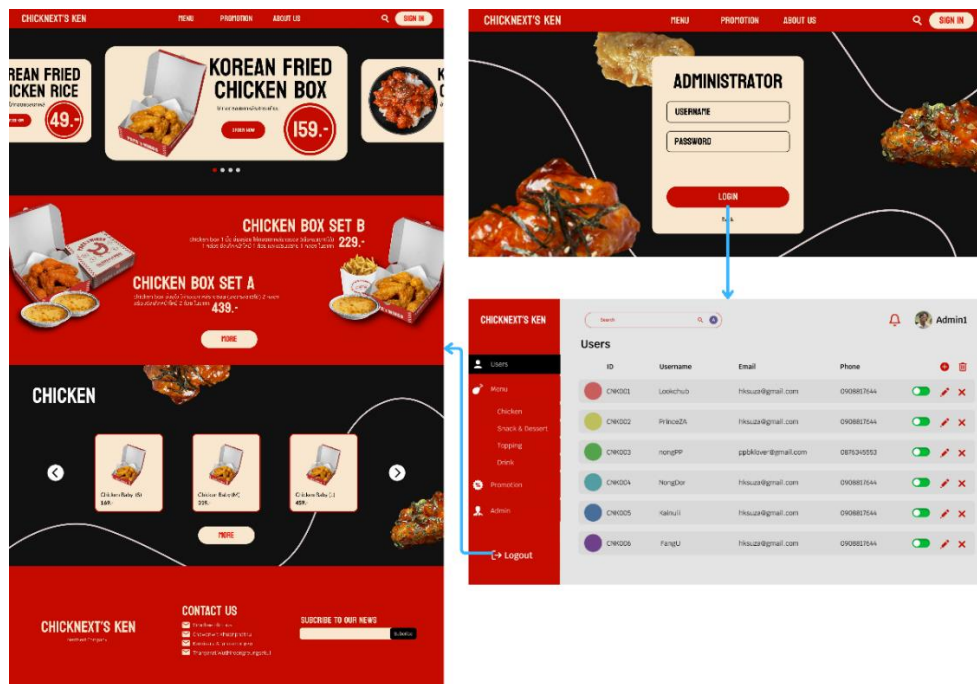


(ภาพที่ 3 แสดงการเชื่อมโยงจากหน้า Home ไปยังหน้าต่างๆ)

2. “Login” Page

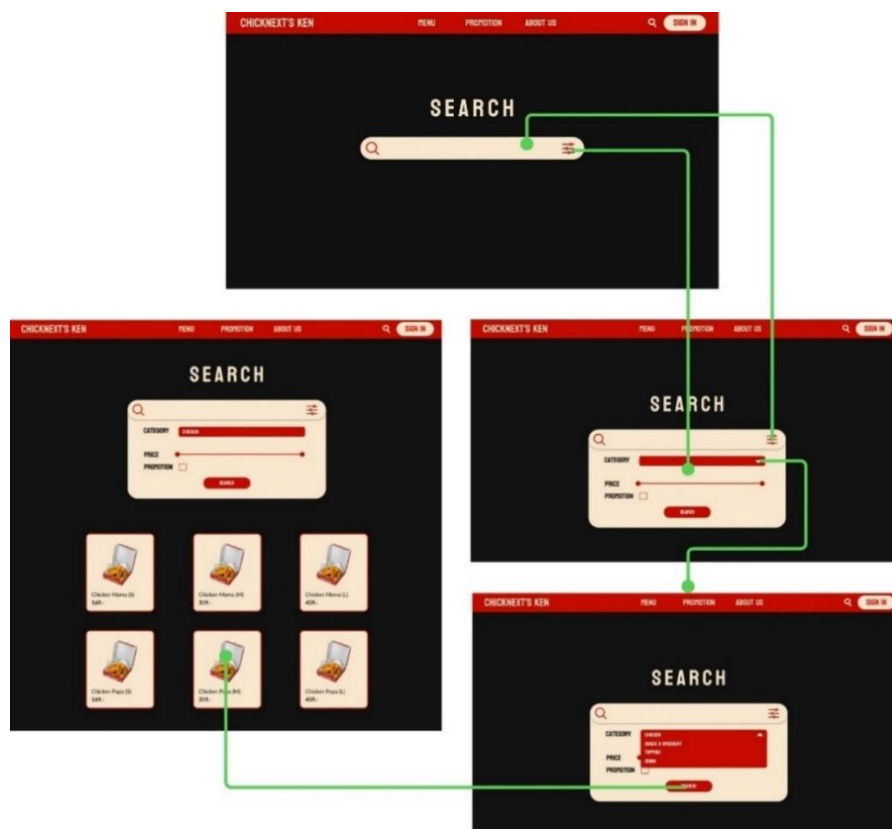


(ภาพที่ 4 แสดงหน้า log-in สำหรับ User และ Administrator และหน้า register สำหรับลูกค้าที่ยังไม่มีสมาชิก)



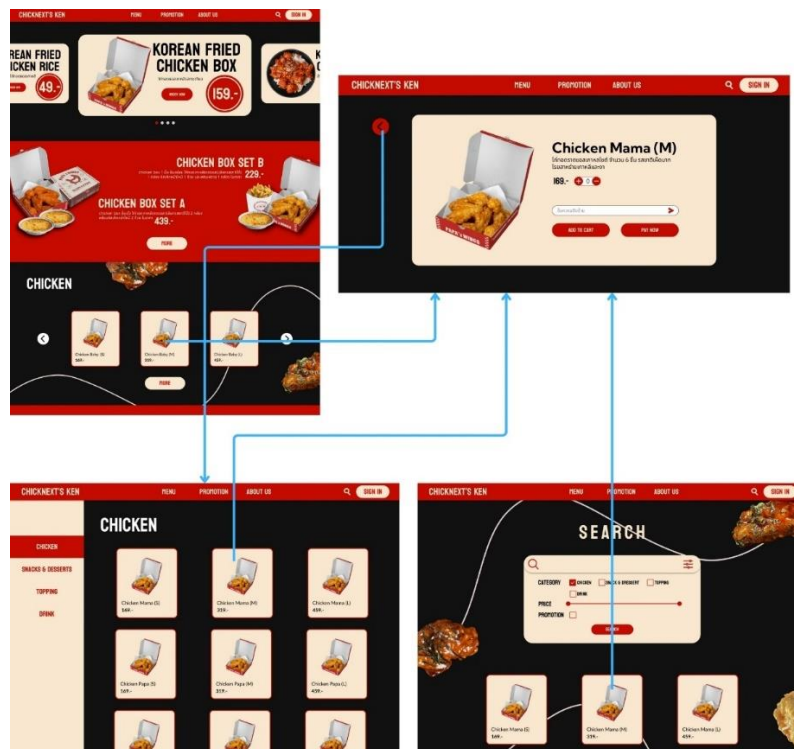
(ภาพที่ 5 แสดงหน้าเมื่อกด login ในหน้า Administrator)

3. “Search” page



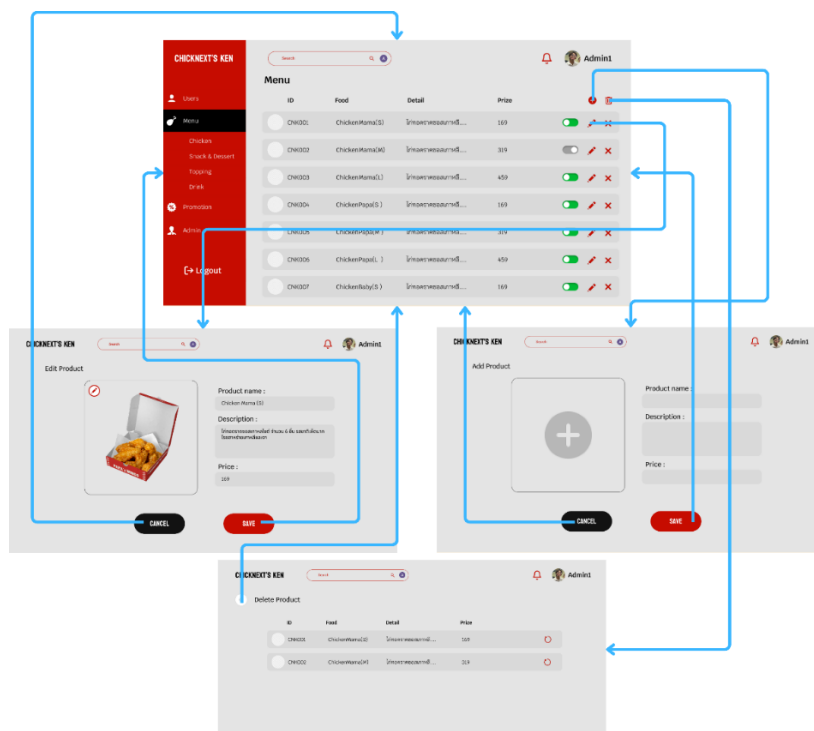
(ภาพที่ 6 แสดงรายละเอียดต่างๆในหน้า Search)

4. “Detail” page



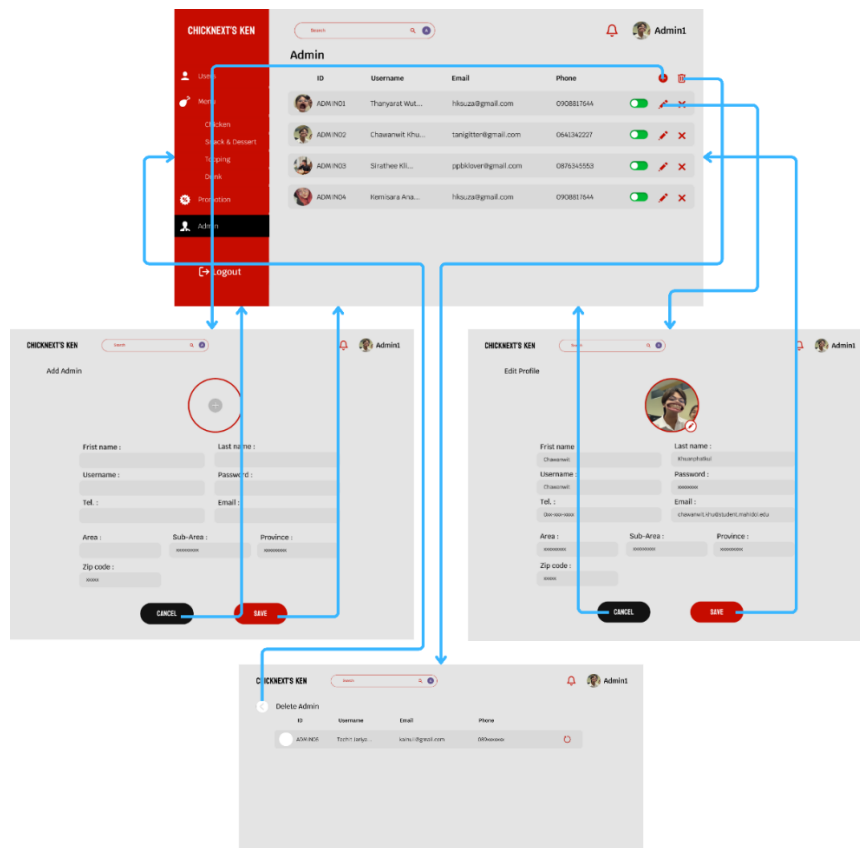
(ภาพที่ 7 แสดงหน้ารายละเอียดของสินค้า)

5. “Product/Service Management” page



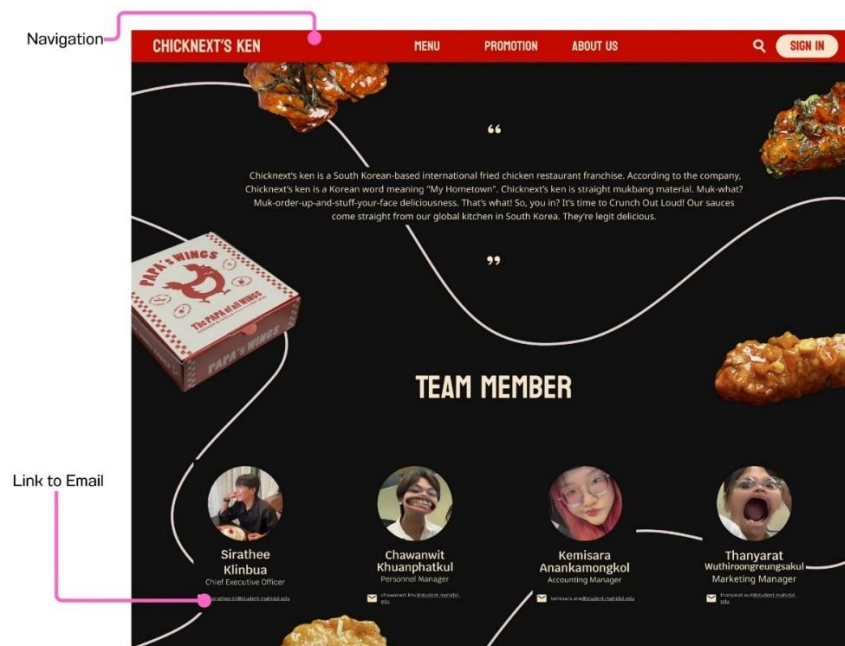
(ภาพที่ 8 แสดงหน้า Product/Service Management ที่สามารถแก้ไขข้อมูลเกี่ยวกับสินค้า)

6. “User Account Management” page



(ภาพที่ 9 หน้า User Account Management ที่สามารถแก้ไขข้อมูลเกี่ยวกับพนักงาน)

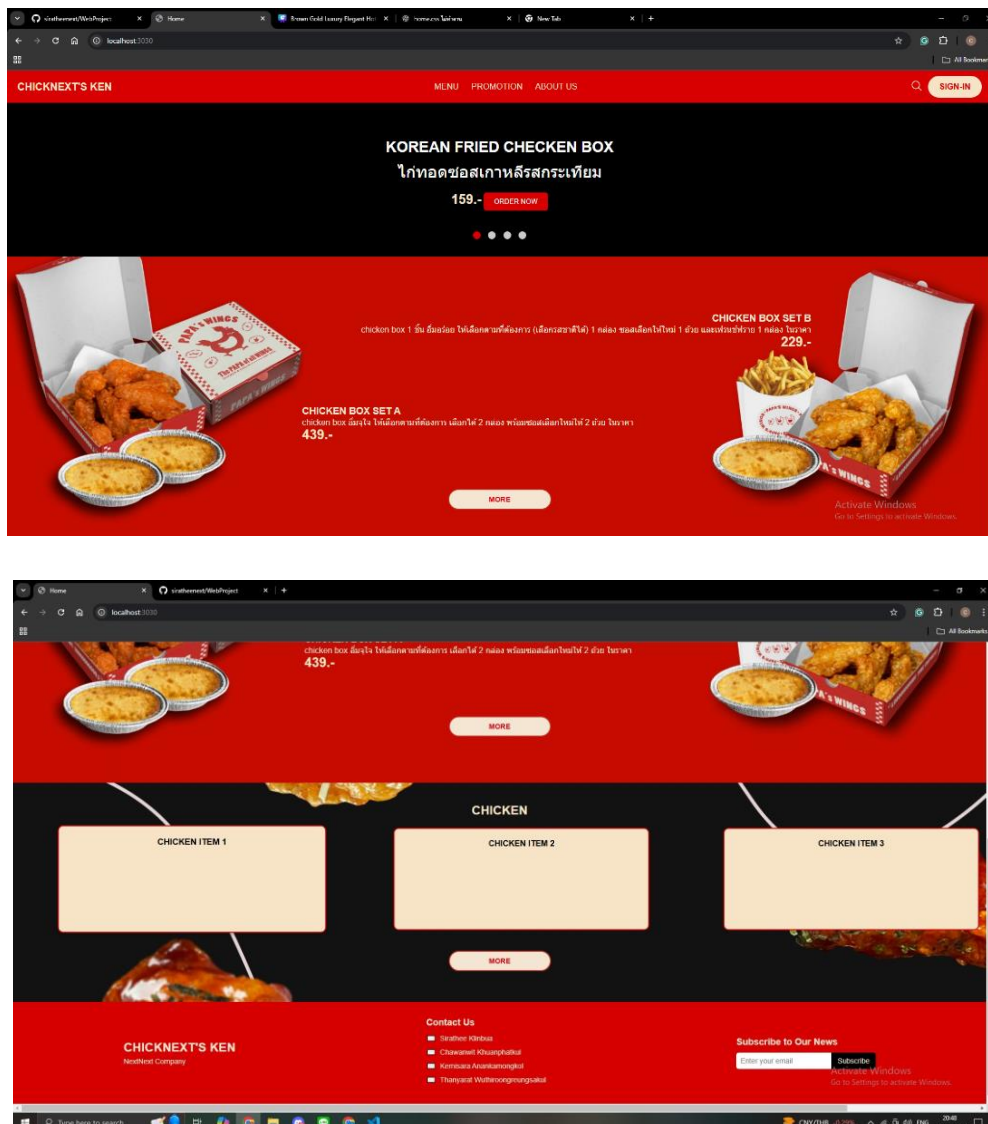
7. “Team” page



(ภาพที่ 10 แสดงหน้า About us ซึ่งเป็นข้อมูลเกี่ยวกับสมาชิกและช่องทางการติดต่อ e-mail)

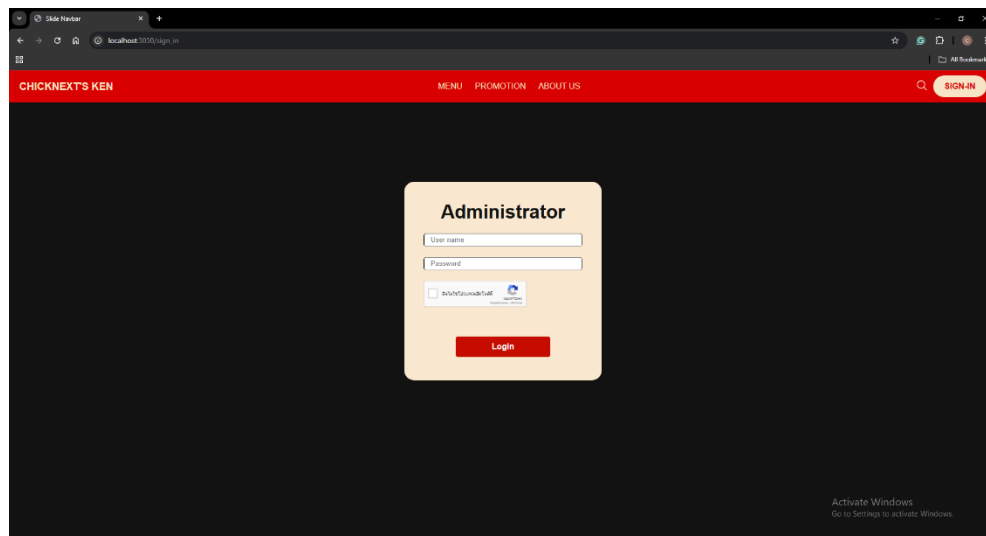
รายละเอียดหน้าเว็บต่างๆ

1. หน้าหลักของเว็บ (Home Page)



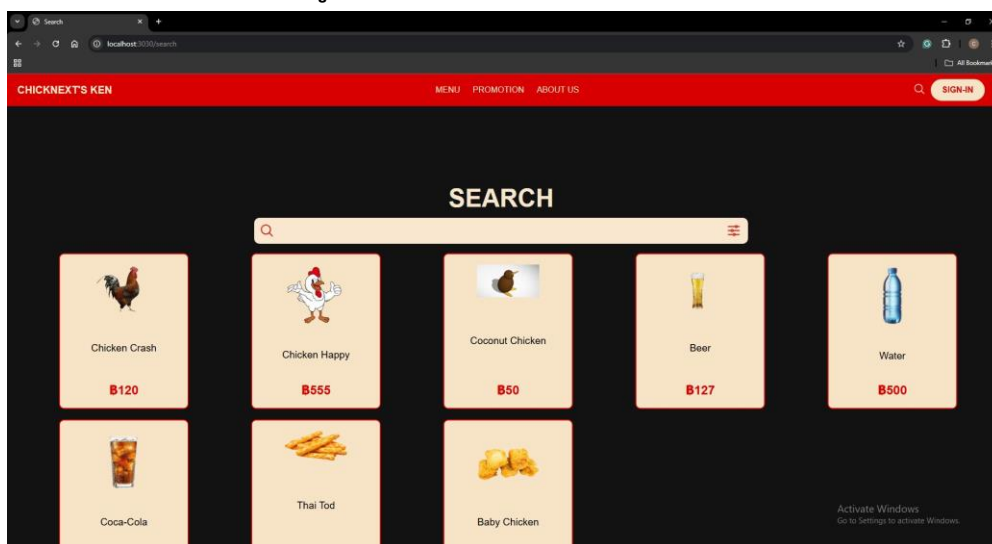
ในหน้าเว็บ Home Page ส่วนบนจะมีแถบ Navigation bar สำหรับการเชื่อมโยงไปยังหน้าอื่นๆ มี Slide show ที่จะแสดงสินค้าแนะนำ โดยในรูปสินค้าแต่ละชิ้นจะมีปุ่ม Order now สำหรับเชื่อมโยงไปยังหน้า Detail ของสินค้านั้นๆ และมีปุ่ม More เพื่อเชื่อมโยงไปยังหน้า Promotion และ Menu ตามลำดับ และส่วนสุดท้ายคือ Footer

2. หน้าเว็บสำหรับการเข้าสู่ระบบ (Log-in) ของผู้ดูแลระบบ



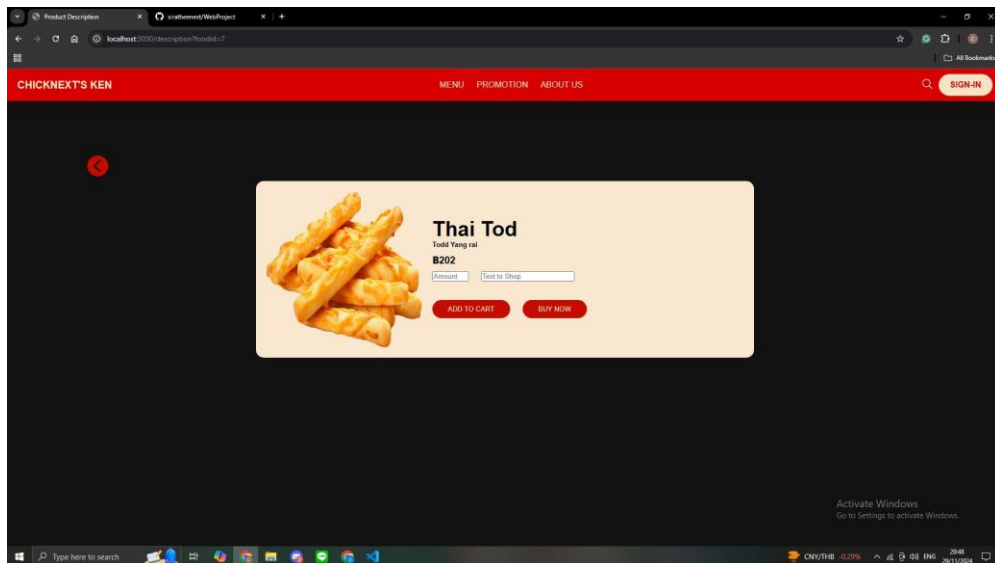
ในส่วนของ admin นั้นจะต้องทำการ login เข้าจากหน้านี้โดยการกดไปที่ปุ่ม SIGN-IN ที่แถบ Navigation bar และทำการกรอก username และ password สำหรับ login เข้าสู่เว็บไซต์ และต้องยืนยันตัวตนผ่าน reCaptcha หาก username และ password ถูกต้องตามฐานข้อมูล myadmin จึงจะสามารถเข้าสู่ระบบได้ ระบบจะทำการบันทึก cookie และบันทึกข้อมูลการเข้าสู่ระบบลงในฐานข้อมูล

3. หน้าสำหรับการค้นหาข้อมูลเว็บในส่วนสินค้าหรือเซอร์วิส



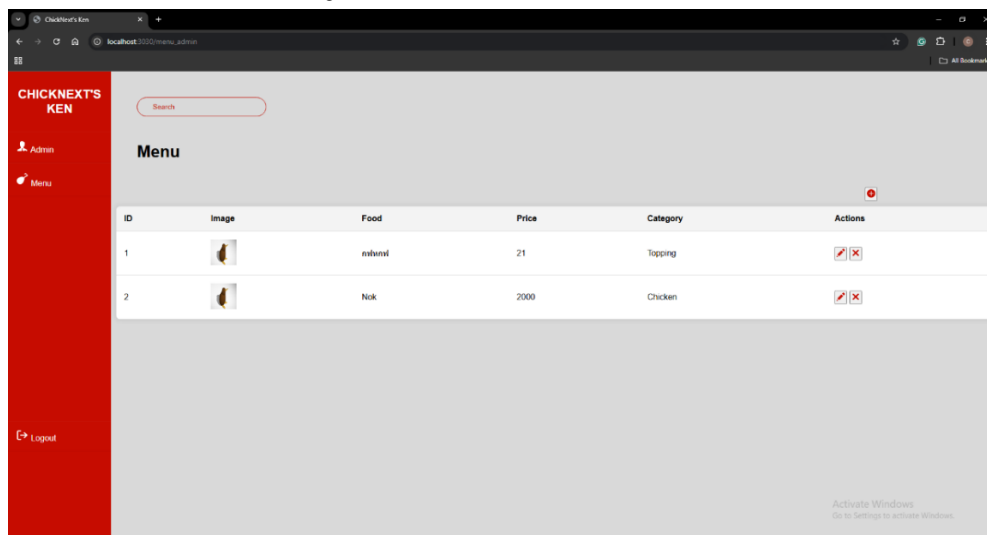
ในหน้า Search จะมีกล่องค้นหาซึ่งสามารถใส่ชื่อรายการอาหารเพื่อค้นหาได้ หรือถ้าหากอยากรจะค้นหาให้ละเอียดกว่านั้นจะมีการกดค้นหาโดยระบุประเภทของอาหาร, ราคา, และยังสามารถตรวจสอบได้ว่าเป็น promotion อยู่หรือไม่ และเมื่อค้นหาอาหารที่ต้องการได้แล้วให้กดไอคอน search เพื่อที่จะแสดงรายการอาหารที่ต้องการค้นหาไว้

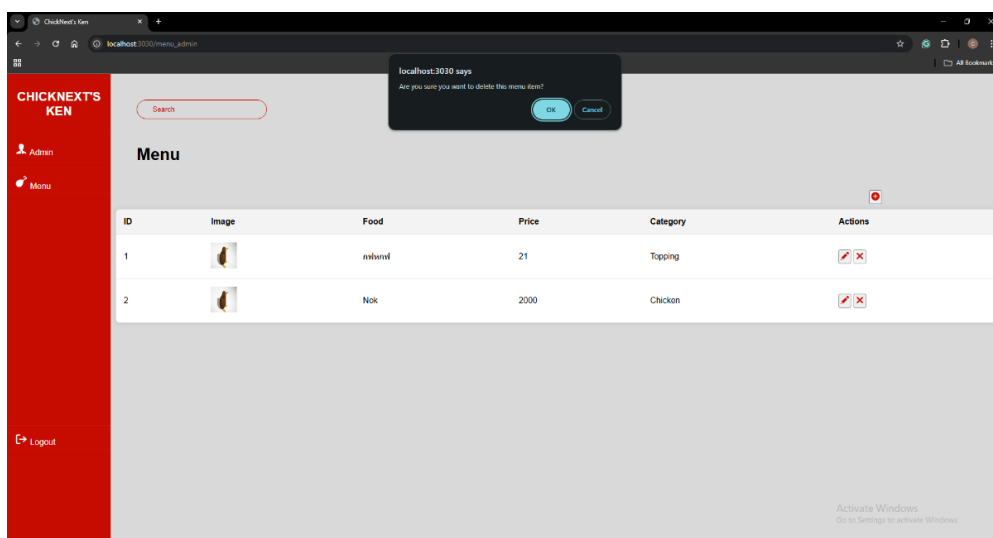
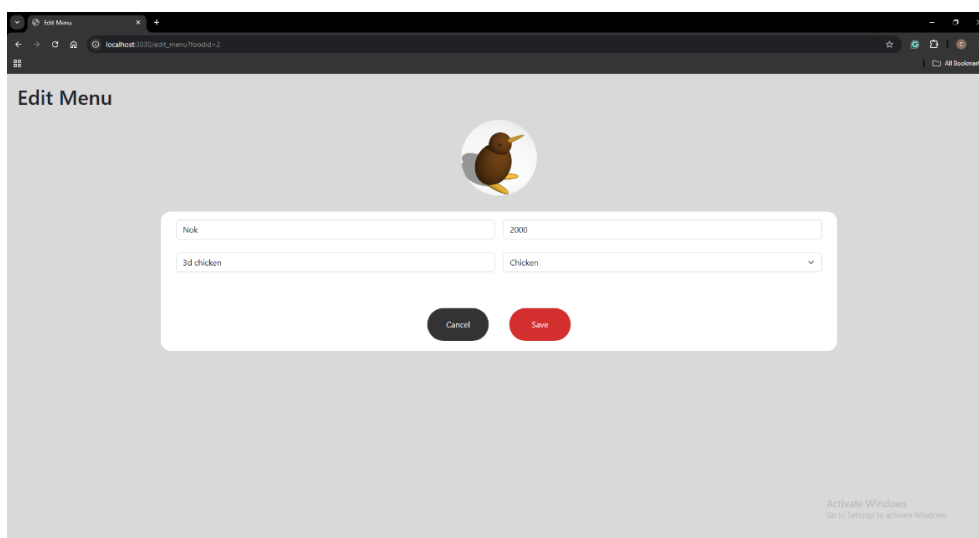
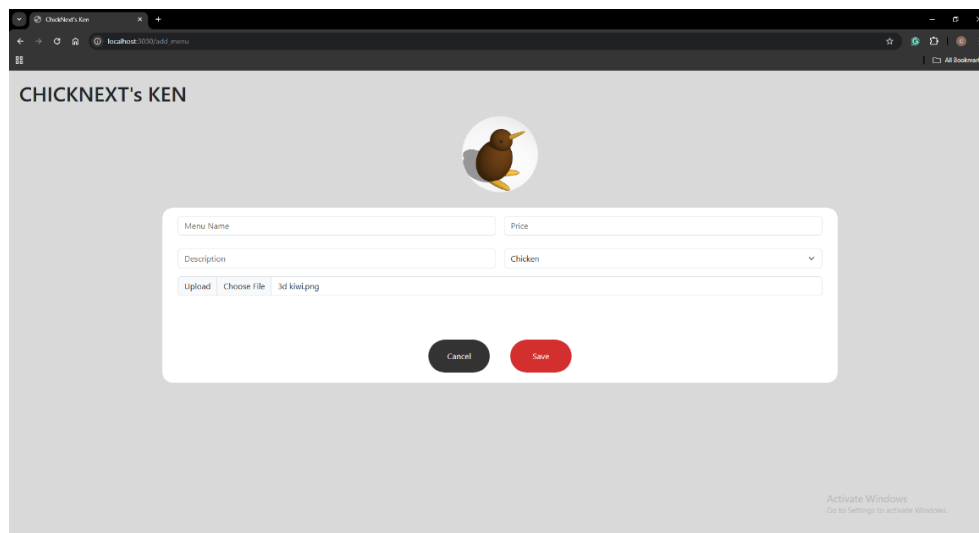
4. หน้าสำหรับการแสดงรายละเอียดสินค้าหรือเซอร์วิส



ในหน้า Detail จะแสดงรายละเอียดของสินค้าเช่น ชื่อสินค้า คำอธิบาย ราคา เป็นต้น การกดเลือกเมนูจะสามารถกดได้จากรูปภาพเมนูโดยตรงแล้วเมื่อนั้นจะทำการเปลี่ยนไปยังอีกหน้าหนึ่ง ซึ่งผู้ใช้สามารถกดย้อนกลับได้และเมื่อกดกลับแล้วจะแสดงหน้าเมนูทั้งหมดที่ร้านมี ณ ตอนนั้น

5. หน้าสำหรับการจัดการข้อมูลของสินค้าหรือเซอร์วิส

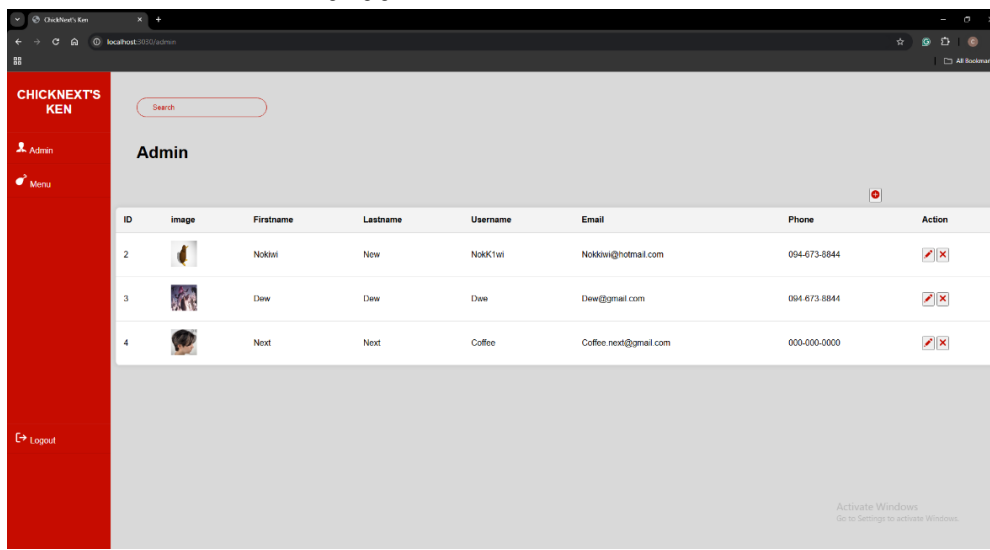













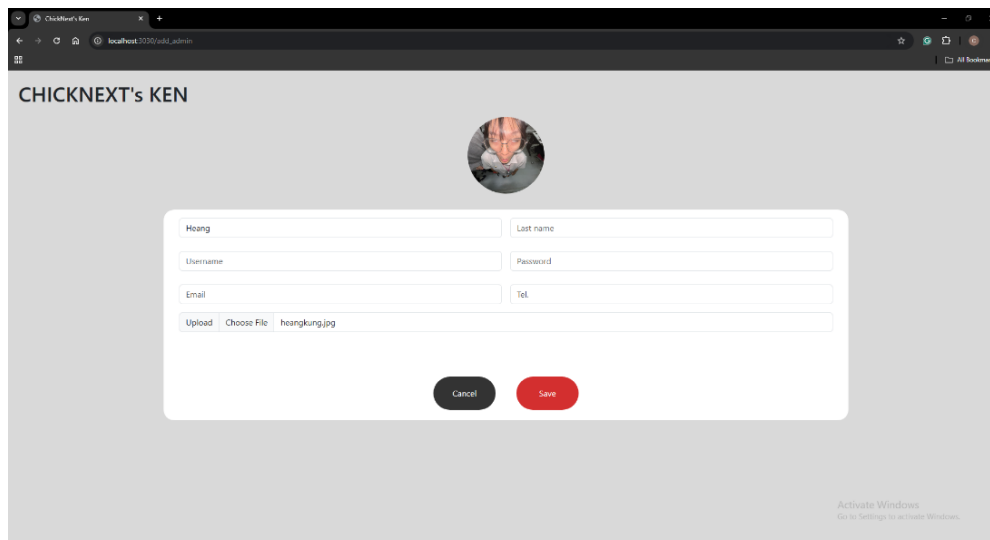
โดยในหน้านี้จะป็นหน้า Product management สามารถดูรายการสินค้าที่มีในร้านสามารถดู ID ชื่อสินค้า หมวดหมู่ รูปภาพ และราคา สามารถแก้ไขข้อมูลสินค้าได้โดยเมื่อกดไอคอนดินสอจะนำไปยังหน้าแก้ไขข้อมูล โดยในหน้าแก้ไขข้อมูลจะสามารถแก้ไขชื่อสินค้า รายละเอียดสินค้า ราคาสินค้า และหมวดหมู่ได้ หากไม่

ต้องการบันทึกข้อมูลและต้องการกลับมายังหน้า Product management สามารถกดปุ่ม cancel ได้และสามารถบันทึกข้อมูลและกลับมายังหน้า Product management ได้โดยการกด save และในหน้า Product management สามารถกดไอคอนกากบาทเพื่อลบสินค้าในกรณีจะไม่ขายสินค้า และสามารถกดไอคอนบวกเพื่อเพิ่มสินค้าได้โดยจะต้องใส่ชื่อสินค้า รายละเอียดสินค้า ราคาสินค้า หมวดหมู่สินค้า และรูปสินค้า หากไม่ต้องการบันทึกข้อมูลและต้องการกลับมายังหน้า Product management สามารถกดปุ่ม cancel ได้และสามารถบันทึกข้อมูลและกลับมายังหน้า Product management ได้โดยการกด save

6. หน้าสำหรับการจัดการข้อมูลผู้ดูแลระบบ



ID	image	Firstname	Lastname	Username	Email	Phone	Action
2		Nokkiwi	New	Nokkiwi	Nokkiwi@hotmail.com	094-673-8844	 
3		Dew	Dew	Dew	Dew@gmail.com	094-673-8844	 
4		Next	Next	Coffee	Coffee.next@gmail.com	000-000-0000	 



CHICKNEXT'S KEN

Hoang

Last name

Username

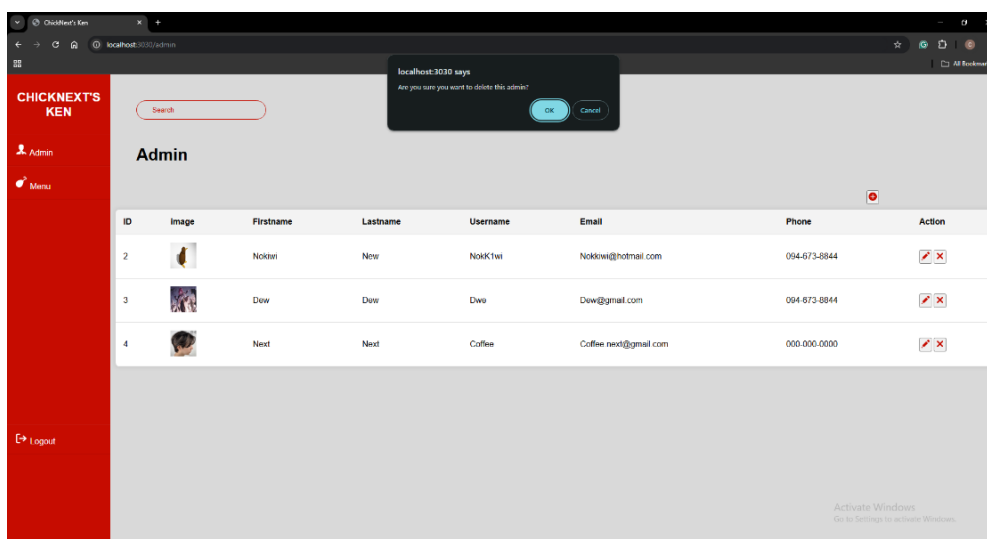
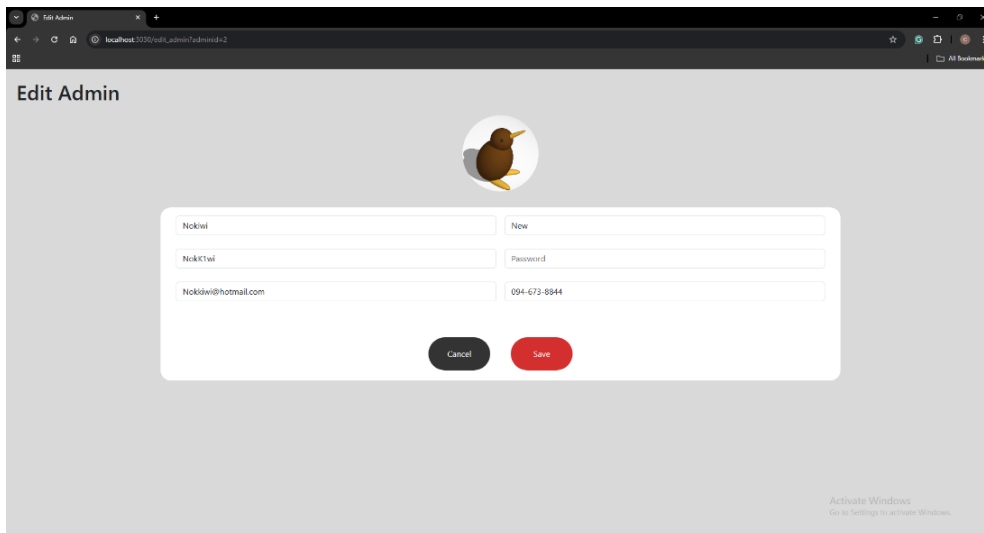
Password

Email

Tel.

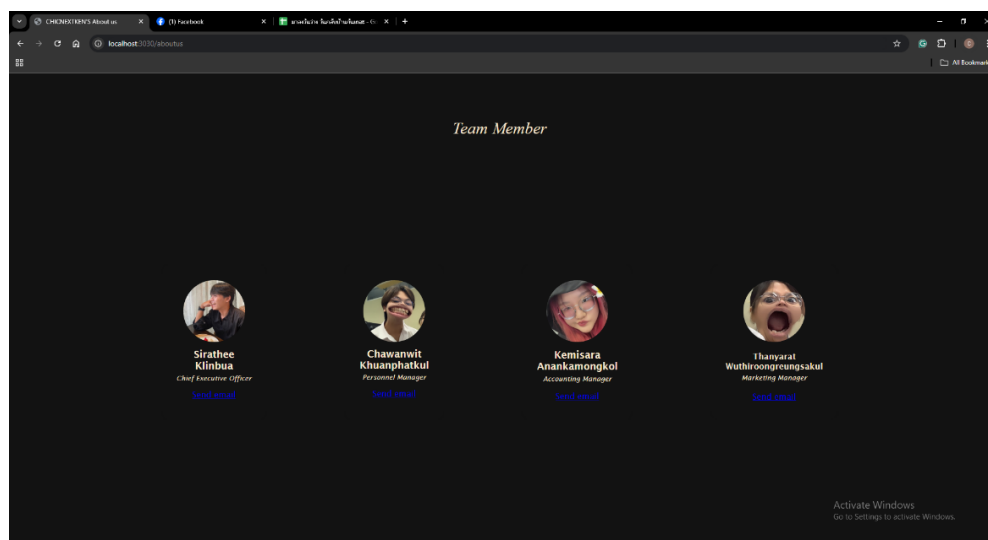
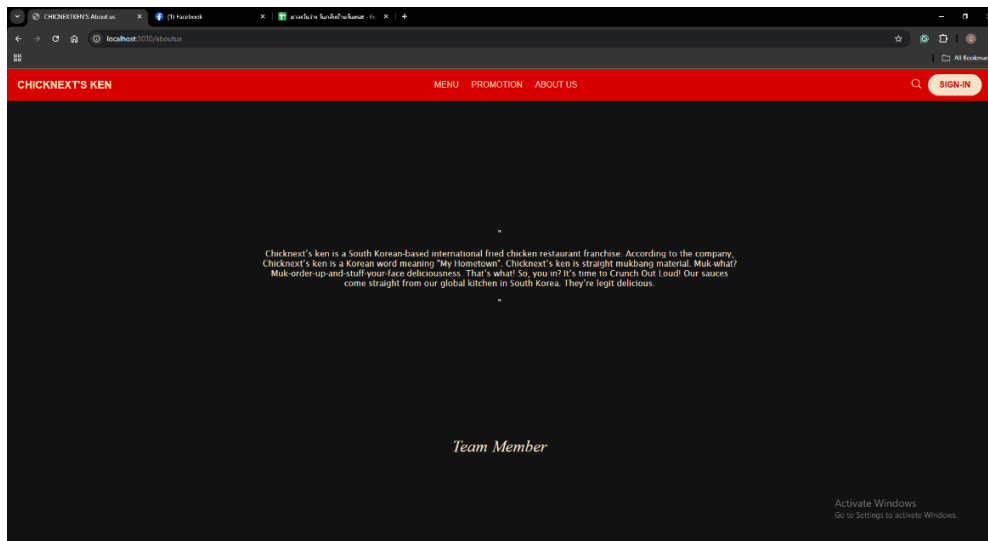
Upload Choose File heangkung.jpg

Cancel Save



โดยในหน้านี้จะป็นหน้า Admin management สามารถดูรายการข้อมูลแอดมินในร้านสามารถดู ID รูปภาพ ชื่อ-นามสกุล username email และเบอร์มีโทรศัพท์ สามารถแก้ไขได้โดยเมื่อกดไอคอนดินสอจะนำไปยังหน้าแก้ไขข้อมูล โดยในหน้าแก้ไขข้อมูลจะสามารถแก้ไขชื่อ-นามสกุล username password email และเบอร์มีโทรศัพท์ หากไม่ต้องการบันทึกข้อมูลและต้องการกลับมายังหน้า Admin management สามารถกดปุ่ม cancel ได้และสามารถบันทึกข้อมูลและกลับมายังหน้า Admin management ได้โดยการกด save และสามารถกดไอคอนกากบาทเพื่อลบบัญชีแอดมิน และสามารถกดไอคอนบวกเพื่อเพิ่มแอดมินได้โดยจะต้องใส่รูปภาพ ชื่อ-นามสกุล username password email และเบอร์มีโทรศัพท์ หากไม่ต้องการบันทึกข้อมูลและต้องการกลับมายังหน้า Product management สามารถกดปุ่ม cancel ได้และสามารถบันทึกข้อมูลและกลับมายังหน้า Product management ได้โดยการกด save

7. หน้า About Us แสดงรายละเอียดสมาชิกในกลุ่ม



โดยในหน้า About us จะมีคำอธิบายร้านโดยสังเขป และมีข้อมูล team member ประกอบด้วยรูป ชื่อ ตำแหน่ง และมีลิงค์อีเมลสำหรับแต่ละบุคคล

รายละเอียดของเว็บเซอร์วิส และโค้ด

1. เว็บเซอร์วิส reCAPTCHA

a. เว็บเซอร์วิส reCAPTCHA (LoginPage.html)

Code

```
<html>
<body>
.
.
.
<div class="login">
  <form action="http://localhost:3031/sign-in" method="POST">
    .
    .
    .
//----- ส่วนที่ 1 : เพิ่ม reCAPTCHA -----//
    <div class="recaptcha-container">
      <div class="g-recaptcha" data-sitekey="6Ldf6YsqAAAAAGiVxajPal-
      HQmPDgtmPyECu-HTX"
      ></div>
    </div>
    <button type="submit" id="log-in">Login</button>
  </form>
</div>
//----- ส่วนที่ 2 : กำหนดการทำงานของ reCAPTCHA -----//
<script>
document.querySelector("form").addEventListener("submit", async (e) => {
  e.preventDefault(); //ไม่ให้รีฟรชหน้า

  const username = document.querySelector("#username").value;
  const password = document.querySelector("#password").value;
  const recaptcha = grecaptcha.getResponse();
```



```

if (!username || !password) {
    swal.fire({
        icon: "error",
        title: "Oops...",
        text: "Please complete all fields",
    });
    return;
}

if (!recaptcha) {
    swal.fire({
        icon: "error",
        title: "Oops...",
        text: "Please complete the reCAPTCHA.",
    });
    return;
}

//----- ส่วนที่ 2 : try catch สำหรับการดักจับข้อผิดพลาด -----//
try {
    const response = await fetch("http://localhost:3031/sign-in", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ username, password, recaptcha }),
        credentials: "include", //ใส่ให้รับคุกกี้ได้
    });

    const result = await response.json();

    if (response.ok) {
        swal.fire({
            title: "Success!",
            text: "Login successful!",
            icon: "success",
        });
    }
}

```

```

        await delay(2000); // ทำอันนี้ก่อนแล้วค่อยทำโค้ดข้างล่าง
        window.location.href = "/admin";

    } else {
        swal.fire({
            icon: "error",
            title: "Oops...",
            text: result.error || "Login failed.",
        });
    }
} catch (err) {
    console.error("Error during login:", err);
    swal.fire({
        icon: "error",
        title: "Oops...",
        text: "An error occurred. Please try again.",
    });
}
});
//----- ส่วนพิเศษสำหรับใช้งาน reCAPTCHA ได้สมบูรณ์ -----//
function delay(ms) {
    return new Promise(resolve => setTimeout(resolve, ms));
}
</script>
</body>
</html>

```

Description

ส่วนที่ 1 กำหนดให้ reCAPTCHA ในส่วนที่ต้องกันในหน้า LoginPage.html ในส่วนของ from

ส่วนที่ 2 กำหนดการทำงานของ reCAPTCHA สร้างฟังก์ชันให้ปุ่ม submit เพื่อให้ปุ่มสามารถใช้งานได้ กำหนดตัวแปร username, password, recaptcha เพื่อเก็บข้อมูล username, password จาก html from และเรียกการทำงานของ reCAPTCHA

เงื่อนไขตรวจสอบการใส่ชื่อผู้ใช้และรหัสผ่าน หากไม่ได้ใส่อย่างใดอย่างหนึ่ง จะแสดงผล pop-up icon "error", title: "Oops...", text: "Please complete all fields"

เงื่อนไขตรวจสอบการยืนยันตัวตนผ่าน reCAPTCHA หากไม่ได้ยืนยัน จะแสดงผล pop-up icon "error", title: "Oops...", text: " Please complete the reCAPTCHA."

Try catch สำหรับการดักจับข้อผิดพลาด กำหนดตัวแปร response เพื่อเก็บข้อมูลสำหรับ fetch (<http://localhost:3031/sign-in>) มีใช้คำสั่ง POST, headers { "Content-Type": "application/json" }, body: JSON.stringify({ username, password, recaptcha }), credentials: "include" เพื่อให้สามารถรับคุกกี้ได้ ต่อมากำหนดตัวแปร result เพื่อเก็บข้อมูล await response.json(); และภายใน try catch มีเงื่อนไข คือ กรณีที่ผู้ใช้งานระบุข้อมูลครบถ้วน และยืนยันตัวตนเรียบร้อยแล้ว จะนำข้อมูลที่ได้ตรวจสอบจากโค้ดในส่วน back.js หากข้อมูลตรงกับฐานข้อมูล จะแสดงผล pop-up icon: "success", title: "Success!", text: "Login successful" และเรียกใช้ฟังก์ชัน delay ก่อนนำไปสู่หน้า admin ส่วนในกรณีที่ข้อมูลที่นำไปตรวจสอบจากโค้ดในส่วน back.js เกิดข้อผิดพลาดใดๆ จะแสดงผล pop-up icon: "error", title: "Oops...", text: ผลลัพธ์ที่ผิดพลาด หรือข้อความ "Login failed." และจะ login ไม่สำเร็จ สุดท้ายในส่วน catch สำหรับดักจับข้อผิดพลาดทั้งหมดนอกเหนือจากนี้ จะแสดงผล pop-up icon: "error", title: "Oops...", text: " An error occurred. Please try again"

ส่วนพิเศษสำหรับใช้งาน reCAPTCHA ได้สมบูรณ์ เป็นฟังก์ชันสำหรับหน่วงเวลา

b. เว็บเซิร์ฟเวอร์ reCAPTCHA (back.js)

Code

```
//----- ส่วนที่ 1 -----//
app.post("/sign-in", async (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  const recaptcha = req.body.recaptcha;

//----- ส่วนที่ 2 -----//
  if (!username || !password) {
    return res.status(400).json({ error: "Username and password are required." });
  }
  if(!recaptcha){
    return res.status(400).json({ error: "Please complete the reCAPTCHA"});
  }

//----- ส่วนที่ 3 : การใช้งาน reCAPTCHA -----//
  const secretKey = "6Ldf6YsqAAAAACVQOpsbUNkLa97Vk7noTCcSjRTR";
```

```

const verificationURL =
`https://www.google.com/recaptcha/api/siteverify?secret=${secretKey}&response=${recaptc
ha}`; //reCaptcha

try {
  const recaptchaResponse = await fetch(verificationURL, { method: "POST" });
  const recaptchaResult = await recaptchaResponse.json();

  if (!recaptchaResult.success) {
    return res.status(400).json({ error: "reCAPTCHA verification failed." });
  }

  //----- ส่วนที่ 3 : การตรวจสอบข้อมูลการเข้าสู่ระบบ -----//
  const sqlQuery = "SELECT * FROM myadmin WHERE username = ?";
  connection.query(sqlQuery, [username], (err, results) => {
    if (err) {
      console.error("Database Error:", err);
      return res.status(500).json({ error: "An error occurred while processing your
request." });
    }

    if (results.length === 0) {
      return res.status(401).json({ error: "Invalid username or password." });
    }

    const admin = results[0]; //เก็บค่า result ตัวแรกที่ได้

    if (password !== admin.mypassword) {
      return res.status(401).json({ error: "Invalid username or password." });
    }

    //----- ส่วนที่ 3 : การส่งคุกกี้เมื่อเข้าสู่ระบบสำเร็จ -----//
    const sessionToken = `cookie-${new Date().getTime()}`;

```

```

const cookieOptions = {
  httpOnly: true,
  maxAge: 24 * 60 * 60 * 1000,
};
res.cookie("cookie", sessionToken, cookieOptions);
res.status(200).json({message: "Login successful", user: {username:admin.username}
});

//----- ส่วนที่ 3 : การบันทึกข้อมูลเข้าสู่ระบบ -----//
const loginTime = new Date().toISOString().slice(0, 19).replace('T', ' ');
const loginQuery = "INSERT INTO log_history (username, pw, saveLogin) VALUES (?,
?, ?)";

connection.query(loginQuery, [admin.username, admin.mypassword, loginTime],
(insertErr) => {
  if (insertErr) {
    console.error("Error inserting login data:", insertErr);
    return res.status(500).json({error: "Failed to record login."});
  }
  console.log(`Login successful by ${username} at ${loginTime}`);

});

});
} catch (err) {
  console.error("Error during reCAPTCHA verification:", err);
  return res.status(500).json({ error: "An error occurred while verifying reCAPTCHA." });
}
});

```

Description

ส่วนที่ 1 เว็บเซิร์ฟเวอร์ของ login ใช้คำสั่ง post และเชื่อมกับ path “/sign-in” มีกำหนดตัวแปร username, password, reCAPTCHA ซึ่งจะดึงข้อมูลแต่ละตัวแปรมาจาก LoginPage.html โดยใช้ body

ส่วนที่ 2 มีการกำหนดเงื่อนไข โดยเงื่อนไขที่ 1 คือ หากไม่ใส่ข้อมูล username และ password จะแสดงผลใน json “Username and password are required.” เงื่อนไขที่ 2 คือ หากไม่ยืนยันตัวตนผ่าน reCAPTCHA จะแสดงผลใน json “Please complete the reCAPTCHA” และ return res.status(400)

ส่วนที่ 3 มีการทำงาน 4 ส่วน คือ การใช้งาน reCAPTCHA การตรวจสอบข้อมูลการเข้าสู่ระบบ การส่งคุกกี้เมื่อเข้าสู่ระบบสำเร็จ และการบันทึกข้อมูลเข้าสู่ระบบ

การใช้งาน reCAPTCHA มีการกำหนดตัวแปร secretKey และ verificationURL เพื่อรับคีย์ของ reCAPTCHA และ API link ตามลำดับ มีการใช้ try, catch เพื่อดักจับข้อผิดพลาด ภายใน try catch มีการ fetch method: "POST" ในตัวแปร recaptchaResponse และใช้งานข้อมูลในรูปแบบ json ในตัวแปร recaptchaResult มีเงื่อนไขหาก reCAPTCHA ไม่สามารถใช้งานได้ จะ return res.status(400) และแสดงผล json “reCAPTCHA verification failed.”

การตรวจสอบข้อมูลการเข้าสู่ระบบ โดยจะเชื่อมกับฐานข้อมูล MYSQL_DATABASE ตาราง myadmin คอลัม username มีการกำหนดตัวแปร sqlQuery เก็บข้อมูลคำสั่ง SELECT * เพื่อเลือกข้อมูลทั้งหมดมาตรวจสอบ มีการกำหนด 3 เงื่อนไข คือ เงื่อนไขที่ 1 สำหรับข้อผิดพลาดหากไม่พบฐานข้อมูล โดยจะแสดงผลพร้อมคำอธิบายของข้อผิดพลาดใน console และ return res.status(500) แสดงผล json “An error occurred while processing your request.” เงื่อนไขที่ 2 สำหรับตรวจสอบชื่อผู้ใช้และรหัสผ่าน เมื่อไม่มีในฐานข้อมูล จะ return res.status(401) และแสดงผล json “Invalid username or password.” จะเก็บค่า results ที่ได้ในตัวแปร admin เพื่อนำไปตรวจสอบใน เงื่อนไขที่ 3 ซึ่งใช้สำหรับตรวจสอบชื่อผู้ใช้และรหัสผ่านว่า ชื่อผู้ใช้นี้ใช้รหัสผ่านนี้หรือไม่ หากไม่ถูกต้องจะ return res.status(401) และแสดงผล json “Invalid username or password.”

การส่งคุกกี้เมื่อเข้าสู่ระบบสำเร็จ กำหนดตัวแปร sessionToken เพื่อเก็บข้อมูล cookie ที่มีข้อมูลเป็น “cookie-วันที่เข้าสู่ระบบ-เวลาที่เข้าสู่ระบบ” กำหนดตัวแปร cookieOptions เพื่อเก็บข้อมูลการป้องกัน cookie และอายุของ cookie ต่อมาจะแสดงผลของ "cookie", sessionToken, cookieOptions ในหน้า Inspect >> Application เมื่อ login สำเร็จ ในเว็บเบราว์เซอร์ และจะ res.status(200) แสดงผล json message: "Login successful", user: {username:admin.username}

การบันทึกข้อมูลเข้าสู่ระบบ กำหนดตัวแปร loginTime เก็บข้อมูลเวลา ณ ตอนเข้าสู่ระบบ รูปแบบ'YYYY-MM-DD HH:MM:SS' กำหนดตัวแปร loginQuery เก็บคำสั่ง “INSERT INTO log_history (username, pw, saveLogin) VALUES (?, ?, ?)” ต่อมาเป็นเงื่อนไขเมื่อ insert ผิดพลาด จะ return

res.status(500) แสดงผล json “Failed to record login.” หากสามารถ insert สำเร็จ จะแสดงผลชื่อที่ผู้ใช้ที่ใช้เข้าสู่ระบบและเวลา ณ ตอนเข้าสู่ระบบ

2. เว็บเซอร์วิส Cookie

a. เว็บเซอร์วิส Cookie (front.js)

Code

```
...
//----- ส่วนที่ 1 : กำหนดตัวแปรสำหรับเรียกใช้ -----
//
const cors = require('cors');
const cookieParser = require("cookie-parser");

app.use(cors());
app.use(cookieParser());
...
//----- ส่วนที่ 2 : ฟังก์ชันเรียกใช้ cookie จาก back.js -----
//
function goin(req, res, next) {
  const cookie = req.cookies['cookie'];
  console.log(cookie);
  if (cookie) {
    console.log("Cookie found:", cookie);
    next();
  } else {
    console.log("No cookie found. Access denied.");
    res.redirect('/');
  }
}
...
//----- ส่วนที่ 3 : เชื่อมพาร์ทที่ต้องการให้ฟังก์ชันทำงาน -----
//
app.get('/admin', goin, (req, res) => {
```


<pre>res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/admin.html')); }); ...</pre>
Description
<p><u>ส่วนที่ 1</u> กำหนดตัวแปรสำหรับเรียกใช้ กำหนดตัวแปร cors สำหรับเรียกใช้ cors กำหนดตัวแปร cookieParser สำหรับเรียกใช้ cookie-parser ต่อมากำหนดให้ app เรียกใช้ cors และ cookie-parser</p> <p><u>ส่วนที่ 2</u> ฟังก์ชันเรียกใช้ cookie จาก back.js โดยมีชื่อฟังก์ชัน คือ goin กำหนดตัวแปร cookie เรียกใช้ req.cookies['cookie'] และแสดงผล cookie ใน console มีการกำหนดเงื่อนไข 2 กรณี คือ กรณีที่ 1 เมื่อเข้าสู่ระบบแล้วซึ่งจะได้รับ cookie หากยังมี cookie อยู่หรือ cookie ยังไม่หมดอายุ จะแสดงผลว่ามี cookie พร้อมบอกรายละเอียดของ cookie นั้นๆ และปล่อยให้สามารถทำ action ต่างๆตามที่กำหนด เช่น การเชื่อมโยงไปยัง path ใดๆ กรณีที่ 2 คือ เมื่อไม่มี cookie หรือ cookie หมดอายุ จะแสดงผล “No cookie found. Access denied.” ใน console และจะเชื่อมโยงมายังหน้าโฮม</p> <p>ส่วนที่ 3 เชื่อมพาร์ทที่ต้องการให้ฟังก์ชันทำงาน กำหนดให้คำสั่ง get เรียกใช้ฟังก์ชัน goin ซึ่งเมื่อเป็นไปตามกรณีที่ 1 จะเชื่อมโยงไปยังหน้า admin</p>

b. เว็บเซอร์วิส Cookie (back.js)

Code
<pre>const sessionToken = `cookie-\${new Date().getTime()}`; const cookieOptions = { httpOnly: true, maxAge: 24 * 60 * 60 * 1000, }; res.cookie("cookie", sessionToken, cookieOptions); res.status(200).json({message: "Login successful", user: {username: admin.username}});</pre>
Description
<p>ส่วนที่กำหนดค่า cookie ในส่วนของโค้ด back.js</p>

3. เว็บเซอร์วิส หน้า Home

a. เว็บเซอร์วิส (Home.html)

Code
<pre>app.get('/', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/Home.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path '/' (root) โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Home.html จากโฟลเดอร์ ../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

4. เว็บเซอร์วิส หน้า Navbar

a. เว็บเซอร์วิส (navbar.html)

Code
<pre>app.get('/navbar', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/navbar.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /navbar โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ navbar.html จากโฟลเดอร์ ../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

5. เว็บเซอร์วิส หน้า Menu

a. เว็บเซอร์วิส (menu.html)

Code
<pre>app.get('/menu', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/menu.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /menu โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ menu.html จากโฟลเดอร์ ../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

b. เว็บเซอวิส Menu (back.js)

Code
<pre>//----- ส่วนที่ 1 -----// app.get('/menu_data', (req, res) => { const sql = 'SELECT * FROM product'; connection.query(sql, (error, results) => { if (error) return res.status(500).send("Error fetching products"); results.forEach(product => { product.imageUrl = product.image ? \${req.protocol}://\${req.get('host')}/uploads/\${product.image} : '/img/default-image.png'; }); res.json(results); }); }); //----- ส่วนที่ 2 -----// app.get('/menu_data/:foodid', (req, res) => { const { foodid } = req.params; const sql = 'SELECT * FROM product WHERE foodid = ?'; connection.query(sql, [foodid], (error, results) => { if (results.length === 0) return res.status(404).send('Product not found'); const product = results[0]; product.imageUrl = product.image ? \${req.protocol}://\${req.get('host')}/uploads/\${product.image} : null; res.send(product); }); });</pre>
Description
<p><u>ส่วนที่ 1</u> Endpoint: /menu_data โดยใช้ HTTP Method: GET การทำงานคือ ดึงข้อมูลสินค้า (menu) ทั้งหมดจากฐานข้อมูล product และเพิ่ม URL สำหรับรูปภาพสินค้าในแต่ละรายการ</p>

ส่วนที่ 2 Endpoint: /menu_data/:foodid โดยใช้ HTTP Method: GET การทำงานคือ ค้นหาสินค้าด้วย foodid และหากไม่พบสินค้าจะส่งสถานะ 404

6. เว็บเซอร์วิส หน้า Promotion

a. เว็บเซอร์วิส (Promotion.html)

Code

```
app.get('/promotion', (req, res) => {  
  res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/Promotion.html'));  
});
```

Description

กำหนด Route สำหรับ HTTP GET ที่ path /promotion โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Promotion.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

7. เว็บเซอร์วิส หน้า About us

a. เว็บเซอร์วิส (AboutusPage.html)

Code

```
app.get('/aboutus', (req, res) => {  
  res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/AboutusPage.html'));  
});
```

Description

กำหนด Route สำหรับ HTTP GET ที่ path /aboutus โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ AboutusPage.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

8. เว็บเซอร์วิส หน้า Search

a. เว็บเซอร์วิส (search.html)

Code
<pre>app.get('/search', (req, res) => { res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/search.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /search โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ search.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

9. เว็บเซอร์วิส หน้า Snack

a. เว็บเซอร์วิส (snack.html)

Code
<pre>app.get('/snack', (req, res) => { res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/snack.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /snack โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ snack.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

10. เว็บเซอร์วิส หน้า Topping

a. เว็บเซอร์วิส (topping.html)

Code
<pre>app.get('/topping', (req, res) => { res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/topping.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /topping โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ topping.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

11. เว็บเซอร์วิส หน้า Drink

a. เว็บเซอร์วิส (drink.html)

Code
<pre>app.get('/drink', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/drink.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /drink โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ drink.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

12. เว็บเซอร์วิส หน้า Description

a. เว็บเซอร์วิส (description.html)

Code
<pre>app.get('/description', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/description.html')); });</pre>
Description
กำหนด Route สำหรับ HTTP GET ที่ path /description โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ description.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

13. เว็บเซอร์วิส หน้า Sign in

a. เว็บเซอร์วิส (LoginPage.html)

Code
<pre>app.get('/sign_in', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/LoginPage.html')); });</pre>
Description

กำหนด Route สำหรับ HTTP GET ที่ path /sign_in โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ LoginPage.html จากโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ กลับไปยัง client เพื่อแสดงหน้าเว็บ

b. เว็บเซอร์วิส Sign in (back.js)

Code

```
app.post("/sign-in", async (req, res) => {  
  const username = req.body.username;  
  const password = req.body.password;  
  
  const sqlQuery = "SELECT * FROM myadmin WHERE username = ?";  
  connection.query(sqlQuery, [username], (err, results) => {  
    if (results.length === 0 || password !== results[0].mypassword) {  
      return res.status(401).json({ error: "Invalid username or password." });  
    }  
    res.status(200).json({ message: "Login successful" });  
  });  
});
```

Description

Endpoint: /sign-in โดยใช้ HTTP Method: POST การทำงานคือ ตรวจสอบ username และ password จากฐานข้อมูล myadmin และส่งข้อความยืนยันหรือแจ้งข้อผิดพลาด

14. เว็บเซอร์วิส หน้า Admin

a. เว็บเซอร์วิส (admin.html)

Code

```
app.get('/admin', goin, (req, res) => {  
  res.sendFile(path.join(__dirname, '../../Sec2_gr8_fe_src/html/admin.html'));  
});
```

Description

เมื่อมีคำขอ GET มาถึง /admin เซิร์ฟเวอร์จะตรวจสอบ Middleware ก่อนดำเนินการและ หากผ่าน Middleware จะส่งไฟล์ admin.html ที่อยู่ในโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ ไปยัง client เพื่อแสดงหน้าเว็บ

b. เว็บเซิร์ฟวิส Admin (back.js)

Code

```
//----- ส่วนที่ 1 -----//
app.get("/admin_data", (req, res) => {
  const sql = `SELECT * FROM myadmin ORDER BY adminid`;
  connection.query(sql, (error, result) => {
    if (error) throw error;
    res.send(result);
  });
});

//----- ส่วนที่ 2 -----//
app.get('/admin_data/:adminid', (req, res) => {
  const { adminid } = req.params;
  const sql = `SELECT * FROM myadmin WHERE adminid = ?`;
  connection.query(sql, [adminid], (error, results) => {
    if (error) return res.status(500).send('Error fetching admin data');
    if (results.length === 0) return res.status(404).send('Admin not found');
    res.send(results[0]);
  });
});
```

Description

ส่วนที่ 1 Endpoint: /admin_data โดยใช้ HTTP Method: GET การทำงานคือ ใช้คำสั่ง SQL SELECT * FROM myadmin ORDER BY adminid; เพื่อดึงข้อมูล Admin ทั้งหมดจากฐานข้อมูลและส่งข้อมูล Admin กลับในรูปแบบ JSON

ส่วนที่ 2 Endpoint: /admin_data/:adminid โดยใช้ HTTP Method: GET การทำงานคือใช้ adminid จาก URL เพื่อค้นหา Admin ในฐานข้อมูลและส่งข้อมูล Admin ที่ตรงกับ ID กลับหรือสถานะ 404 หากไม่พบข้อมูล

15. เว็บเซอร์วิส หน้า Sidebar

a. เว็บเซอร์วิส (sidebar.html)

Code
<pre>app.get('/sidebar', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/sidebar.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /sidebar เซิร์ฟเวอร์จะส่งไฟล์ sidebar.html ที่อยู่ในโฟลเดอร์ ../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดง Sidebar บนหน้าเว็บ

16. เว็บเซอร์วิส หน้า Menu_admin

a. เว็บเซอร์วิส (menu_admin.html)

Code
<pre>app.get('/menu_admin', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/menu_admin.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /menu_admin เซิร์ฟเวอร์จะส่งไฟล์ menu_admin.html ที่อยู่ในโฟลเดอร์ ../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดงหน้าเว็บจัดการเมนูของระบบ

17. เว็บเซอร์วิส หน้า Add_admin

a. เว็บเซอร์วิส (add_admin.html)

Code
<pre>app.get('/add_admin', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/add_admin.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /add_admin เซิร์ฟเวอร์จะส่งไฟล์ add_admin.html ที่อยู่ในโฟลเดอร์ ../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดงหน้าเว็บสำหรับเพิ่มข้อมูล Admin ใหม่ในระบบ

b. เว็บเซอร์วิส Add_admin (back.js)

Code
<pre>app.post("/add_admin", upload.single('profileImage'), (req, res) => { const { firstName, lastName, username, password, email, tel } = req.body; const profileImage = req.file ? req.file.filename : null; const sql = `INSERT INTO myadmin (Fname, Lname, username, mypassword, email, tel, profile_image) VALUES (?, ?, ?, ?, ?, ?, ?)`; connection.query(sql, [firstName, lastName, username, password, email, tel, profileImage], (error) => { if (error) return res.status(500).send("Error inserting admin"); res.redirect("http://localhost:3030/admin"); }); });</pre>
Description
Endpoint: /add_admin โดยใช้ HTTP Method: POST การทำงานคือ รับข้อมูล Admin ใหม่จาก req.body และรูปภาพจาก req.file เพิ่มข้อมูลลงในฐานข้อมูลด้วยคำสั่ง SQL INSERT INTO

18. เว็บเซอร์วิส หน้า Add_menu

a. เว็บเซอร์วิส (add_menu.html)

Code
<pre>app.get('/add_menu', (req, res) => { res.sendFile(path.join(__dirname, '../..../Sec2_gr8_fe_src/html/add_menu.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /add_menu เซิร์ฟเวอร์จะส่งไฟล์ add_menu.html ที่อยู่ในโฟลเดอร์ ../../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดงหน้าเว็บเพื่อเพิ่มข้อมูลเมนูใหม่ในระบบ

19. เว็บเซิร์ฟวิส หน้า Edit_admin

a. เว็บเซิร์ฟวิส (edit_admin.html)

Code
<pre>app.get('/edit_admin', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/edit_admin.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /edit_admin เซิร์ฟเวอร์จะส่งไฟล์ edit_admin.html ที่อยู่ในโฟลเดอร์ ../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดงหน้าเว็บแก้ไขข้อมูล Admin

b. เว็บเซิร์ฟวิส Edit_admin (back.js)

Code
<pre>//----- ส่วนที่ 1 -----// app.delete("/admin_data/:adminid", (req, res) => { const { adminid } = req.params; const getImageSql = `SELECT profile_image FROM myadmin WHERE adminid = ?`; connection.query(getImageSql, [adminid], (error, results) => { if (results.length === 0) return res.status(404).send("Admin not found"); const imageName = results[0]?.profile_image; const deleteSql = `DELETE FROM myadmin WHERE adminid = ?`; connection.query(deleteSql, [adminid], (deleteError) => { if (imageName) { const imagePath = path.join(__dirname, "uploads", imageName); fs.unlink(imagePath, () => console.log(`Image deleted: \${imagePath}`)); } res.send("Admin deleted successfully"); }); }); });</pre>

```
});
//----- ส่วนที่ 2 -----//
app.put("/admin_data/:adminid", upload.single('profileImage'), (req, res) => {
  const { adminid } = req.params;
  const { firstName, lastName, username, password, email, tel } = req.body;
  const profileImage = req.file ? req.file.filename : null;

  let sql = `UPDATE myadmin SET Fname = ?, Lname = ?, username = ?, mypassword = ?,
email = ?, tel = ? WHERE adminid = ?`;
  const values = [firstName, lastName, username, password, email, tel, adminid];

  if (profileImage) {
    sql = `UPDATE myadmin SET Fname = ?, Lname = ?, username = ?, mypassword = ?,
email = ?, tel = ?, profile_image = ? WHERE adminid = ?`;
    values.splice(6, 0, profileImage);
  }

  connection.query(sql, values, (error) => {
    if (error) return res.status(500).send("Error updating admin");
    res.send("Admin updated successfully");
  });
});
```

Description

ส่วนที่ 1 Endpoint: /admin_data/:adminid โดยใช้ HTTP Method: DELETE การทำงานคือ ลบ ข้อมูล Admin และรูปโปรไฟล์ที่เกี่ยวข้อง

ส่วนที่ 2 Endpoint /admin_data/:adminid โดยใช้ HTTP Method: PUT การทำงานคือ อัปเดต ข้อมูล Admin พร้อมรองรับการอัปเดตโปรไฟล์รูปภาพ

20. เว็บเซอร์วิส หน้า Edit_menu

c. เว็บเซอร์วิส (edit_menu.html)

Code
<pre>app.get('/edit_menu', (req, res) => { res.sendFile(path.join(__dirname, '../Sec2_gr8_fe_src/html/edit_menu.html')); });</pre>
Description
เมื่อมีคำขอ GET มายัง /edit_menu เซิร์ฟเวอร์จะส่งไฟล์ edit_menu.html ที่อยู่ในโฟลเดอร์ ../Sec2_gr8_fe_src/html/ ไปยัง client ใช้สำหรับแสดงหน้าเว็บแก้ไขข้อมูลเมนูในระบบ

d. เว็บเซอร์วิส Edit_menu (back.js)

Code
<pre>//----- ส่วนที่ 1 -----// app.get('/menu_data', (req, res) => { const sql = 'SELECT * FROM product'; connection.query(sql, (error, results) => { results.forEach(product => { product.imageUrl = product.image ? `\${req.protocol}://\${req.get('host')}/uploads/\${product.image}` : '/img/default-image.png'; }); res.json(results); }); }); //----- ส่วนที่ 2 -----// app.post("/add_menu", upload.single('menuImage'), (req, res) => { const { menuName, price, description, category } = req.body; const menuImage = req.file ? req.file.filename : null; const sql = `INSERT INTO product (foodname, detail, price, category, image) VALUES (?, ?, ?, ?, ?)`;</pre>

```

    connection.query(sql, [menuName, description, price, category, menuImage], (error) =>
    {
        if (error) return res.status(500).send("Error adding product");
        res.redirect("http://localhost:3030/menu_admin");
    });
});

//----- ส่วนที่ 3 -----//

app.delete("/delete_menu/:foodid", (req, res) => {
    const { foodid } = req.params;

    const getImageSql = `SELECT image FROM product WHERE foodid = ?`;
    connection.query(getImageSql, [foodid], (error, results) => {
        const imageName = results[0]?.image;
        const deleteSql = `DELETE FROM product WHERE foodid = ?`;

        connection.query(deleteSql, [foodid], (deleteError) => {
            if (imageName) {
                const imagePath = path.join(__dirname, "uploads", imageName);
                fs.unlink(imagePath, () => console.log(`Image deleted: ${imagePath}`));
            }
            res.send("Product deleted successfully");
        });
    });
});

//----- ส่วนที่ 4 -----//

app.put("/edit_menu/:foodid", upload.single('menuImage'), (req, res) => {
    const { foodid } = req.params;
    const { menuName, price, description, category } = req.body;
    const menuImage = req.file ? req.file.filename : null;

```



```

let sql, values;

if (!menuImage) {
    sql = `UPDATE product SET foodname = ?, detail = ?, price = ?, category = ? WHERE
foodid = ?`;
    values = [menuName, description, price, category, foodid];
} else {
    sql = `UPDATE product SET foodname = ?, detail = ?, price = ?, category = ?, image =
? WHERE foodid = ?`;
    values = [menuName, description, price, category, menuImage, foodid];
}

connection.query(sql, values, (error) => {
    if (error) return res.status(500).send("Error updating product");
    res.send("Product updated successfully");
});
});

```

Description

ส่วนที่ 1 Endpoint: /menu_data โดยใช้ HTTP Method: GET การทำงานคือ ดึงข้อมูลสินค้า และสร้าง URL ของรูปภาพสำหรับการแสดงผล

ส่วนที่ 2 Endpoint: /add_menu โดยใช้ HTTP Method: POST การทำงานคือ เพิ่มสินค้าใหม่ในฐานข้อมูล พร้อมรูปภาพ (ถ้ามี)

ส่วนที่ 3 Endpoint: /delete_menu/:foodid โดยใช้ HTTP Method: DELETE การทำงานคือ ลบข้อมูลสินค้าและไฟล์รูปภาพที่เกี่ยวข้อง

ส่วนที่ 4 Endpoint /edit_menu/:foodid โดยใช้ HTTP Method: PUT การทำงานคือ อัปเดตข้อมูลสินค้า พร้อมเปลี่ยนรูปภาพใหม่ (ถ้ามี)

ผลการทดสอบของเว็บเซิร์ฟเวอร์ทั้งหมด

URL	http://localhost:3031/admin_data
Method	GET
Body	<pre>[{ "adminid": 1, "Fname": "New", "Lname": "Kiwi", "username": "NokKiwi", "mypassword": "566359", "email": "Nokkiwi@kiwi.com", "tel": "055-555-5555", "profile_image": "1732833206882-kiwi run.jpg" }, { "adminid": 2, "Fname": "Dew", "Lname": "Nongten", "username": "NongDor", "mypassword": "123333", "email": "Lovetenmark@dew.com", "tel": "010-101-1001", "profile_image": "1732833269680-mem03.jpg" }, { "adminid": 3, "Fname": "Heang", "Lname": "Kung", "username": "Heangkung", "mypassword": "HHHHHH123", "email": "Heang.kung@heang.com", "tel": "083-333-3333",</pre>

	<pre> "profile_image": "1732833304191-heangkung.jpg" }, { "adminid": 4, "Fname": "Next", "Lname": "Coffee", "username": "AraiKordai", "mypassword": "adaikorra", "email": "Next.Coffee@Cafeien.com", "tel": "000-000-0000", "profile_image": "1732833399045-mem04.jpg" }] </pre>
Description	ดึงข้อมูล admin ทั้งหมด

URL	http://localhost:3031/admin_data/1
Method	GET
Body	<pre> { "adminid": 1, "Fname": "New", "Lname": "Kiwi", "username": "NokKiwi", "mypassword": "566359", "email": "Nokkiwi@kiwi.com", "tel": "055-555-5555", "profile_image": "1732833206882-kiwi run.jpg" } </pre>
Description	ดึงข้อมูล admin ที่มี id=1

URL	http://localhost:3031/menu_data
Method	GET
Body	<pre>[{ "foodid": 1, "foodname": "Beer", "detail": "ฟองฟู เมาๆกันั้นจะจี๋ะ", "price": 127, "category": "Drink", "image": "1732833601763-1018.png_860-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833601763-1018.png_860-removebg-preview.png" }, { "foodid": 2, "foodname": "Water", "detail": "It TiME tO dRinK wAtER", "price": 500, "category": "Drink", "image": "1732833662301-pngtree-water-bottle-png-image_13275849-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833662301-pngtree-water-bottle-png-image_13275849-removebg-preview.png" }, { "foodid": 3, "foodname": "Coca-Cola", "detail": "สามารถกินแทนน้ำเปล่าได้", "price": 10, "category": "Drink", "image": "1732833704085-pngtree-drink-recipes-png-image_13442920-removebg-preview.png",</pre>

	<pre> "imageUrl": "http://localhost:3031/uploads/1732833704085-pngtree- drink-recipes-png-image_13442920-removebg-preview.png" }, { "foodid": 4, "foodname": "Chicken Crash", "detail": "หงอนสว่างเงางาม น่องที่แรง 120 แรงม้า วิ่งได้ 300 Km/Hr", "price": 1999, "category": "Chicken", "image": "1732833862541-Kai_Chon-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833862541-Kai_Chon- removebg-preview.png" }, { "foodid": 5, "foodname": "Chicken Happy", "detail": "กินแล้วเป็นไข้ไก่อารมณ์ดี ปลอดสารเสพติด 100%", "price": 555, "category": "Chicken", "image": "1732833921651-CH_Kram_Toh-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833921651- CH_Kram_Toh-removebg-preview.png" }, { "foodid": 6, "foodname": "Coconut Chicken", "detail": "This is chicken not Nokkiwi", "price": 26, "category": "Chicken", "image": "1732833978928-3d kiwi.png", "imageUrl": "http://localhost:3031/uploads/1732833978928-3d kiwi.png" }, }, </pre>
--	---

	<pre> { "foodid": 7, "foodname": "Thai Tod", "detail": "มันทอดไ้", "price": 48, "category": "Snack&Desserts", "image": "1732834144502-che_sss.jpg", "imageUrl": "http://localhost:3031/uploads/1732834144502- che_sss.jpg" }, { "foodid": 8, "foodname": "Baby Chicken", "detail": "เป็นลูกของ Happy Chicken", "price": 55, "category": "Snack&Desserts", "image": "1732834183719-snack_nut_sa_ket-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732834183719- snack_nut_sa_ket-removebg-preview.png" }, { "foodid": 9, "foodname": "Kiwi", "detail": "หวานๆ สดจากฟาร์ม", "price": 96, "category": "Topping", "image": "1732834240043-kiwi.jpg", "imageUrl": "http://localhost:3031/uploads/1732834240043-kiwi.jpg" }, { "foodid": 10, "foodname": "Banana", "detail": " Bello! Me want banana! Tulaliloo ti amo! ", </pre>
--	--

	<pre> "price": 11, "category": "Topping", "image": "1732834510283-pngegg.png", "imageUrl": "http://localhost:3031/uploads/1732834510283- pngegg.png" }] </pre>
Description	ค้นหาข้อมูล product ทั้งหมด

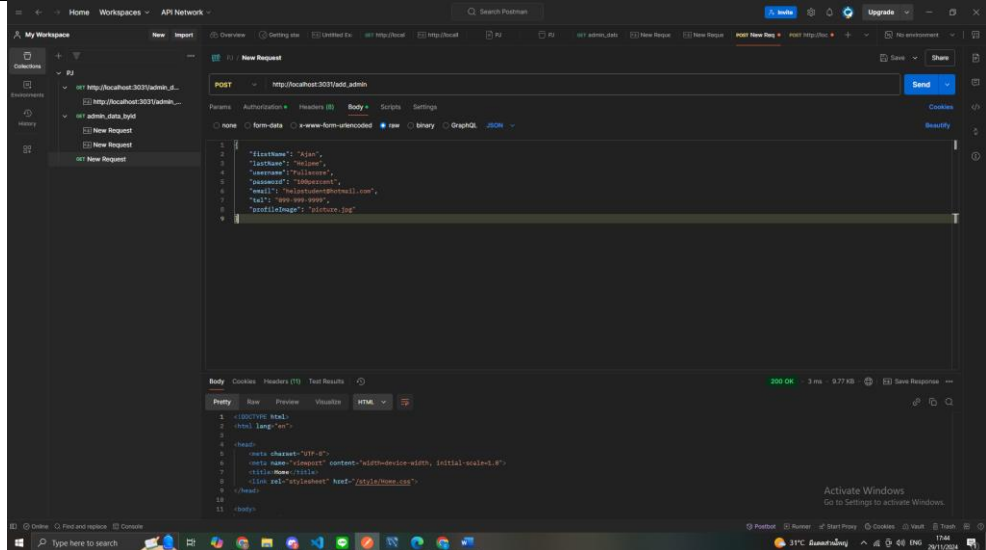
URL	http://localhost:3031/menu_data/1
Method	GET
Body	<pre> { "foodid": 1, "foodname": "Beer", "detail": "ฟองฟู เมากันนะจ๊ะ", "price": 127, "category": "Drink", "image": "1732833601763-1018.png_860-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833601763-1018.png_860- removebg-preview.png" } </pre>
Description	ดึงข้อมูล products ที่มี id=1

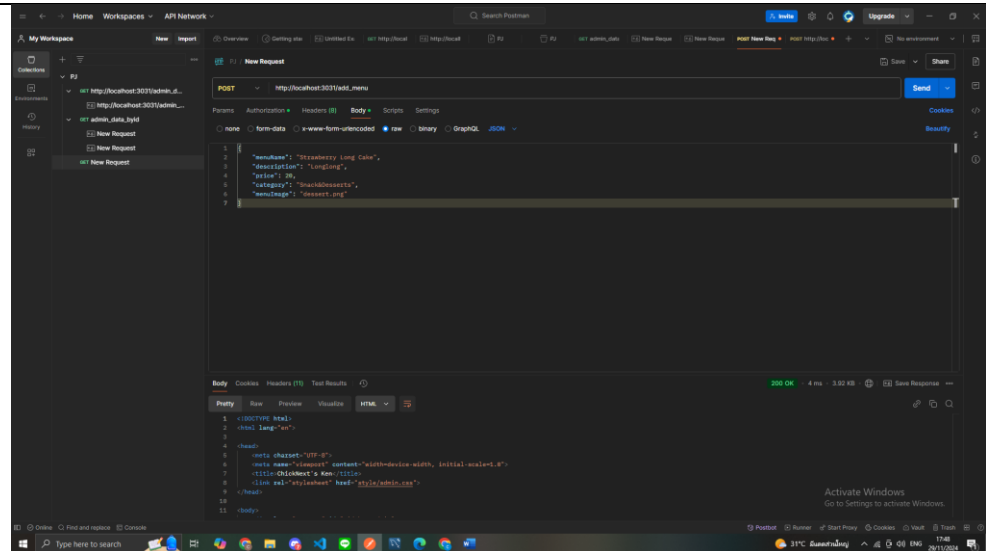
URL	http://localhost:3031/menu_data/category/Chicken
Method	GET
Body	<pre> [{ "foodid": 4, "foodname": "Chicken Crash", "detail": "หงอนสว่างเงางาม น่องที่แรง 120 แรงม้า วิ่งได้ 300 Km/Hr", "price": 1999, </pre>

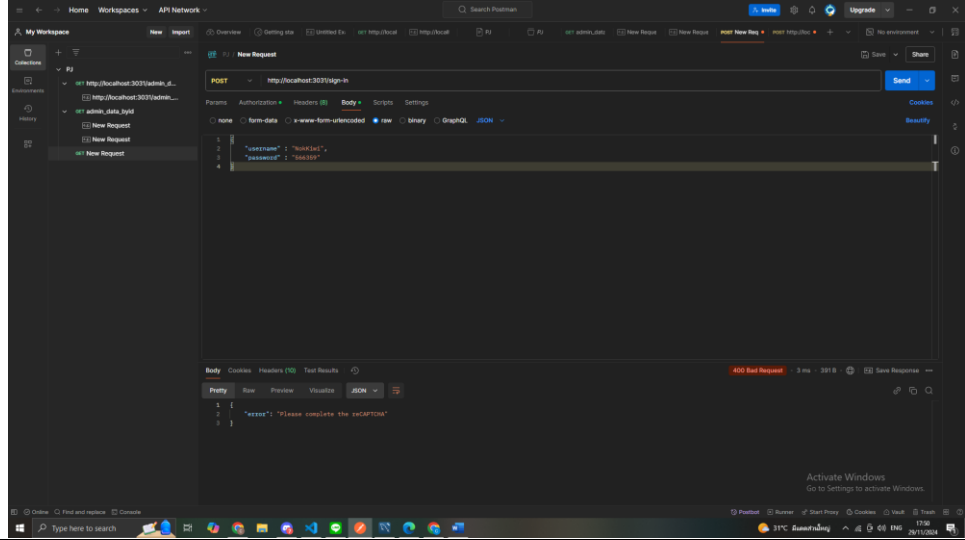
	<pre> "category": "Chicken", "image": "1732833862541-Kai_Chon-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833862541-Kai_Chon- removebg-preview.png" }, { "foodid": 5, "foodname": "Chicken Happy", "detail": "กินแล้วเป็นไข้ไก่อารมณ์ดี ปลอดสารเสพติด 100%", "price": 555, "category": "Chicken", "image": "1732833921651-CH_Kram_Toh-removebg-preview.png", "imageUrl": "http://localhost:3031/uploads/1732833921651- CH_Kram_Toh-removebg-preview.png" }, { "foodid": 6, "foodname": "Coconut Chicken", "detail": "This is chicken not Nokkiwi", "price": 26, "category": "Chicken", "image": "1732833978928-3d kiwi.png", "imageUrl": "http://localhost:3031/uploads/1732833978928-3d kiwi.png" }] </pre>
Description	ดึงข้อมูล product ที่มี attribute category=Chicken

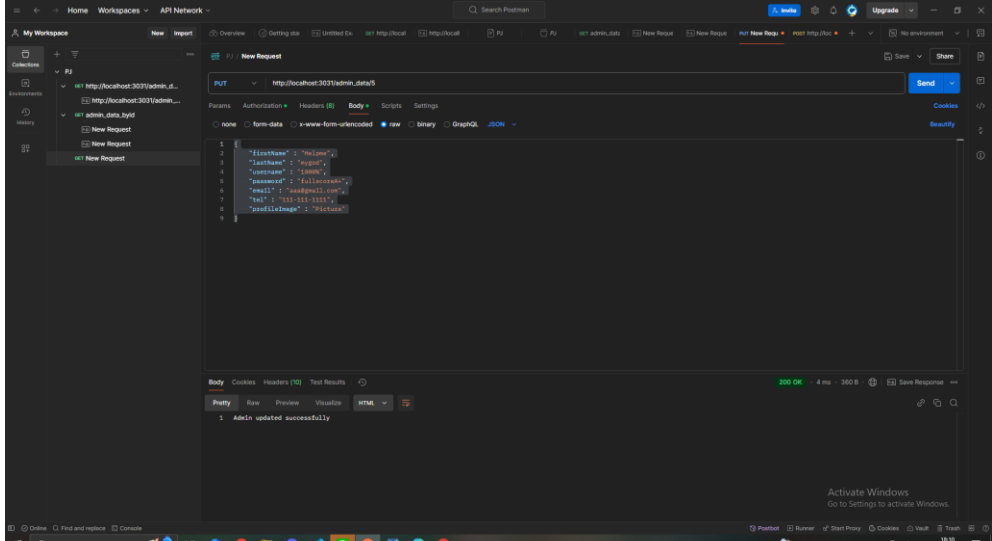
URL	http://localhost:3031/menu_data/products
Method	GET
Body	none

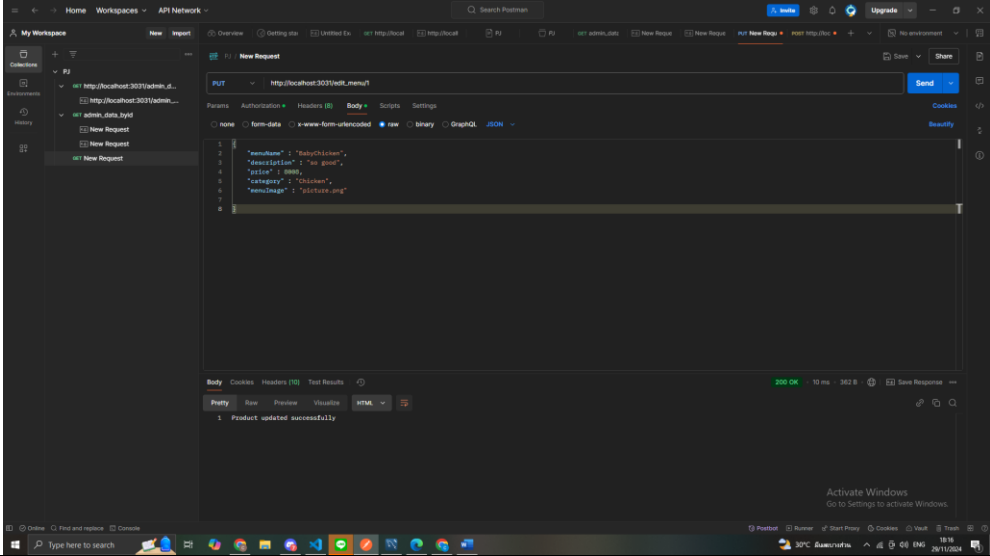
Description	ดึงข้อมูล product ที่ได้รับค่ามาเช่น attribute foodname="Beer" ก็จะดึงข้อมูลนั้นๆ ออกมา
-------------	---

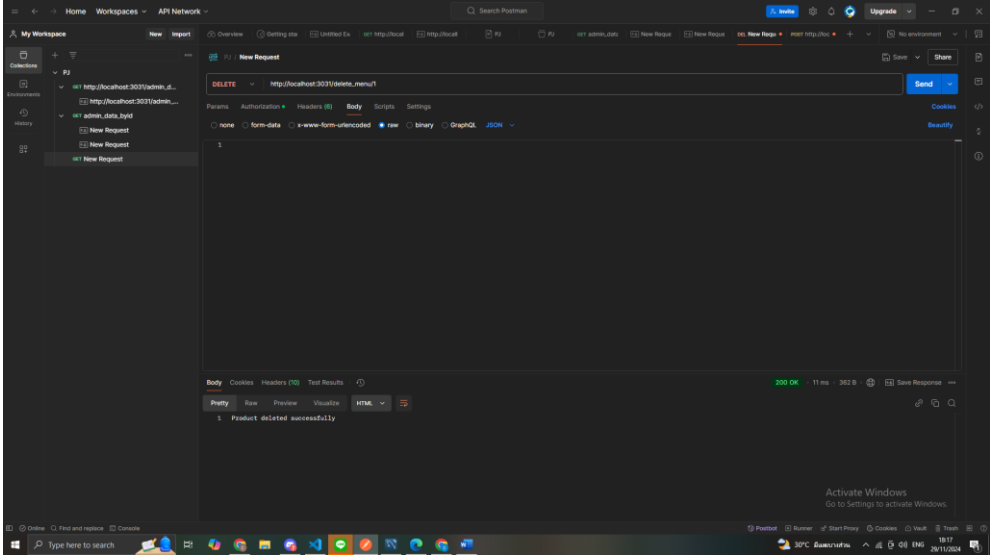
URL	http://localhost:3031/add_admin
Method	POST
Body	
Description	นำข้อมูล insert ไปในตาราง myadmin

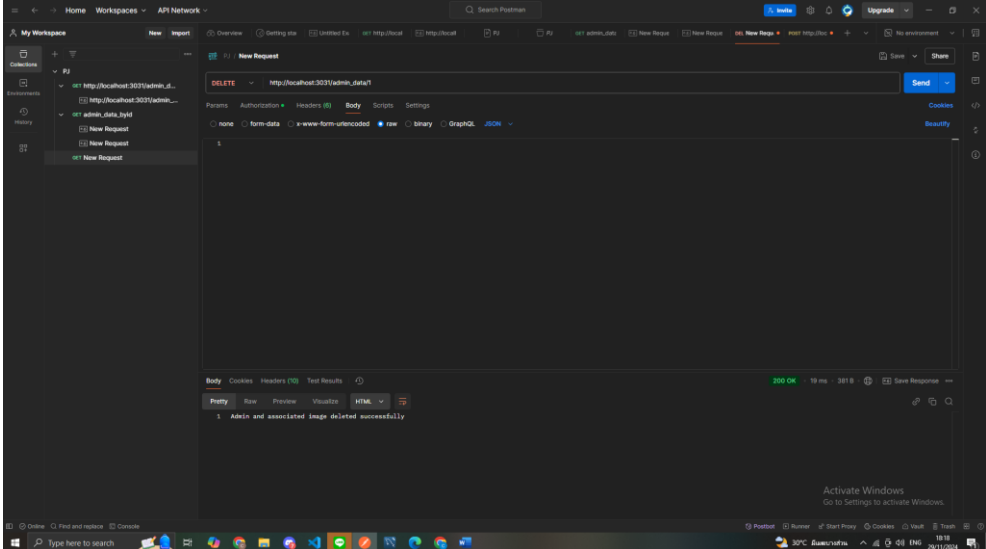
URL	http://localhost:3031/add_menu
Method	POST
Body	
Description	นำข้อมูล insert ไปยังตาราง product

URL	http://localhost:3031/sign-in
Method	POST
Body	
Description	นำ username และ password ไป ค้นหาในตาราง myadmin ทำ captcha และเมื่อผ่านแล้วจะนำข้อมูล username password และ วันเวลาที่ login ใส่ง่ายตาราง log_history

URL	http://localhost:3031/admin_data/5
Method	PUT
Body	
Description	เปลี่ยนข้อมูลของ admin ที่มี id = 5 โดยของกลุ่มพวกเราใช้ method get เขามาร่วม ด้วยในการดึงข้อมูลเก่า จึงต้องใส่ข้อมูลทุกช่อง

URL	http://localhost:3031/edit_menu/1
Method	PUT
Body	
Description	เปลี่ยนข้อมูลของ product ที่มี id = 5 โดยของกลุ่มพวกเราใช้ method put เขามาร่วมด้วยในการดึงข้อมูลเก่า จึงต้องใส่ข้อมูลทุกช่อง

URL	http://localhost:3031/delete_menu/1
Method	DELETE
Body	
Description	ลบข้อมูลในตาราง product ที่มี food id = 1

URL	http://localhost:3031/delete_admin/1
Method	DELETE
Body	
Description	ลบข้อมูลในตาราง myadmin ที่มี adminid = 1

แหล่งอ้างอิง

อ้างอิงข้อมูลจากเว็บไซต์

- [Bonchon Thailand](#) และ [KFC Thailand](#)
- [reCAPTCHA](#)
- [ChatGPT](#)