



รายวิชา 060243110 JavaScript Programming

จัดทำโดย

1.นาย พันธวัล	เมี้ยนศิริ	6206021621040(หัวหน้ากลุ่ม)
2.นาย วรเมธ	วงศาสวัสดิ์	6206021621163
3.นาย จตุรภัทร	จันทร์สีดา	6206021611095
4.นาย กิตติพัฒน์	คำสงค์	6206021611150
5.นาย สิทธิชัย	โสภา	6206021621074

นักศึกษาชั้นปีที่2 ตอนเรียนที่ 2

เสนอ

รองศาสตราจารย์ อนิราช มิ่งขวัญ

ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม ปี การศึกษา 2563 ภาคเรียนที่ 2

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ วิทยาเขตปทุมธานี

## คำนำ

รายงานเล่มนี้เป็นรายละเอียดการจัดทำระบบ เว็บไซต์อาหารล่วงหน้า ส่วนหนึ่งของวิชา 060243110 JavaScript Programming จัดทำขึ้นเพื่อแสดงศึกษาการเขียนโปรแกรมภาษาจาวาสคริป ในวิชาเรียน ซึ่งเป็นการนำความรู้จากการเรียนภาคปฏิบัติในห้องเรียนมาประยุกต์ใช้ เพื่อให้เกิดความรู้ความ เข้าใจในการเรียนมากขึ้น โดยในรายงานเล่มนี้จะประกอบไปด้วย รายละเอียดของระบบงาน การออกแบบระบบงาน การออกฐานข้อมูลเพื่อใช้งานร่วมกันและออกแบบหน้าตาของโปรแกรม เพื่อมาประยุกต์ใช้

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า รายงานเล่มนี้จะก่อให้เกิดประโยชน์แก่ผู้ที่สนใจศึกษาในเรื่อง การเขียน โปรแกรมภาษาจาวาสคริปไม่มากนักน้อย แล้วถ้าหากมีข้อผิดพลาดประการใด ทางผู้จัดทำ ต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

## สารบัญ

โปรแกรมระบบเว็บสั่งอาหารล่วงหน้า	4
ซอฟต์แวร์ที่เกี่ยวข้องในการจัดทำโปรแกรม	5
Source Code	11
บรรณานุกรม	43

## โปรแกรมระบบเว็บสั่งอาหารล่วงหน้า

### ชื่อโปรแกรม

ระบบเว็บสั่งอาหารล่วงหน้า

### จัดทำโดย

1.นาย พันธวัล	เมี้ยนศิริ	6206021621040 (หัวหน้ากลุ่ม)
2.นาย วรเมธ	วงศาสวัสดิ์	6206021621163
3.นาย จตุรภัทร	จันทร์สีดา	6206021611095
4.นาย กิตติพัฒน์	คำสงค์	6206021611150
5.นาย สิทธิชัย	โสภา	6206021621074

นักศึกษาชั้นปีที่ 2 ตอนเรียนที่ 2

### รายละเอียดโปรแกรม

เป็นโปรแกรมในการจัดการเกี่ยวกับการเข้าเว็บเพื่อสั่งอาหารจากร้านอาหารใต้หอชายล่วงหน้าและสามารถลงมารับอาหารตามที่สั่งได้เลยเมื่อเสร็จสิ้นแล้ว

### ที่มาและความสำคัญของโปรแกรม

โปรแกรมระบบเว็บสั่งอาหารล่วงหน้าทางผู้จัดทำได้จัดทำขึ้นเพื่อสร้างความรู้ความเข้าใจในภาษาของจาวาสคริปต์ที่ได้จากการเขียนภาคทฤษฎีและภาคปฏิบัติของการเขียนโปรแกรมในห้องเรียน โดยมีการเขียนโปรแกรม ออกมาเป็นหน้าเว็บเพจ อีกทั้งยังสามารถเชื่อมต่อกับฐานข้อมูลต่างๆได้โดยใช้โปรแกรม VS Code(Visual Studio Code)

## ซอฟต์แวร์ที่เกี่ยวข้องในการจัดทำโปรแกรม

- MongoDB Atlas

MongoDB Atlas นี้จะช่วยให้การเริ่มต้นใช้งาน, การกำหนดค่า, การดูแลรักษา และการปรับแต่งต่างๆ เป็นไปแบบ Automate ทั้งหมดจากศูนย์กลาง ทำให้ไม่ต้องมีผู้ดูแลระบบมากอย่างแต่ก่อน ในขณะที่รองรับความสามารถทางด้านการรักษาความปลอดภัยได้ในหลากหลายรูปแบบที่อ้างอิงจาก AWS ไม่ว่าจะเป็นการทำ Access Control, การทำ Network Isolation ด้วย Amazon VPC, การทำ IP Whitelist, การเข้ารหัสการรับส่งข้อมูลด้วย TLS/SSL และการเข้ารหัสที่ระดับ Filesystem

- Vue.js

คือมันทำหน้าที่เป็น View ใน MVC (Model View Controller) นั่นแหละ เป็น JavaScript Framework ที่พัฒนาโดย Evan You เอาไว้สำหรับพัฒนาพวก UI (User Interface) และในบาง Framework เช่น Laravel ก็ใช้ Vue เป็น Template สำหรับส่วน UI (Frontend) ซึ่ง Vue.js ไม่ได้มี back ใหญ่ๆแบบ Angular (Google) หรือ React (Facebook) แต่ก็มี community ชาวจีนที่ค่อนข้างใหญ่ รวมถึง Alibaba ด้วย

- Node.js

Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript สรุปรวมๆ NodeJS ก็คือ Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server นั่นเอง

- POSTMAN

Postman คือเครื่องมือสำหรับช่วยในการพัฒนา API ทดสอบการทำงานของ Service รวมถึงการ Mock Service อีกด้วย ซึ่งช่วยเราทำเรื่องยาก ๆ ให้กลายเป็นเรื่องง่าย โดยความนิยมของ Postman หลัก ๆ มาจาก UI ที่สวยงามใช้งานง่ายกว่า Tools อื่น ๆ โดยผู้ใช้งานไม่จำเป็นต้องมีความรู้เรื่องภาษาโปรแกรมมิ่งก็สามารถใช้งานได้อย่างสบายใจ

- VS Code

VS Code หรือ Visual Studio Code จากบริษัทไมโครซอฟต์ เป็นโปรแกรมประเภท Editor ใช้ในการแก้ไขโค้ดที่มีขนาดเล็ก แต่มีประสิทธิภาพสูง เป็น OpenSource โปรแกรมจึงสามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานหลายแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows , macOS และ Linux รองรับหลายภาษาทั้ง JavaScript, TypeScript และ Node.js ในตัว และสามารถเชื่อมต่อกับ Git ได้ง่าย สามารถนำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือและส่วนขยายต่าง ๆ ให้เลือกใช้มากมาย รองรับการใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++ , C# , Java , Python , PHP หรือ Go สามารถปรับเปลี่ยน Themes ได้ มีส่วน Debugger และ Commands เป็นต้น ซึ่งบทความนี้จะเป็นการสอน วิธีการใช้งาน Visual Studio Code เบื้องต้น

- GitHub

มันก็คือเว็บไซต์ที่ให้บริการ Git (version control repository) นั่นเอง โดย Github จะให้บริการบนออนไลน์แพลตฟอร์ม ทำให้คุณสามารถเข้าถึงข้อมูลผ่านหน้าเว็บไซต์ได้ทุกที่ทุกเวลา ในส่วนของการใช้บริการก็มีให้ใช้ทั้งแบบเสียเงินและไม่เสียเงิน หากต้องการใช้บริการแบบฟรี ๆ Code ในโปรเจกต์ของเราทั้งหมด คนอื่นจะสามารถเข้าถึงได้ แต่ถ้าอยากใช้แบบส่วนตัวทาง GitHub ก็มี Plan รองรับในราคาค่าใช้จ่ายที่ไม่สูงมาก

## ขั้นตอนการออกแบบระบบ

หน้าแรกเป็นล็อกอิน มีรายละเอียดดังนี้

- มีที่กรอกข้อมูลการ log-in (เข้าสู่ระบบ)
- มีที่กรอกข้อมูลเพื่อ sign-in (สมัครสมาชิก)

# ร้านข้าวไต้หอ


username

scooter

password

12345

go



username

password

Password 2

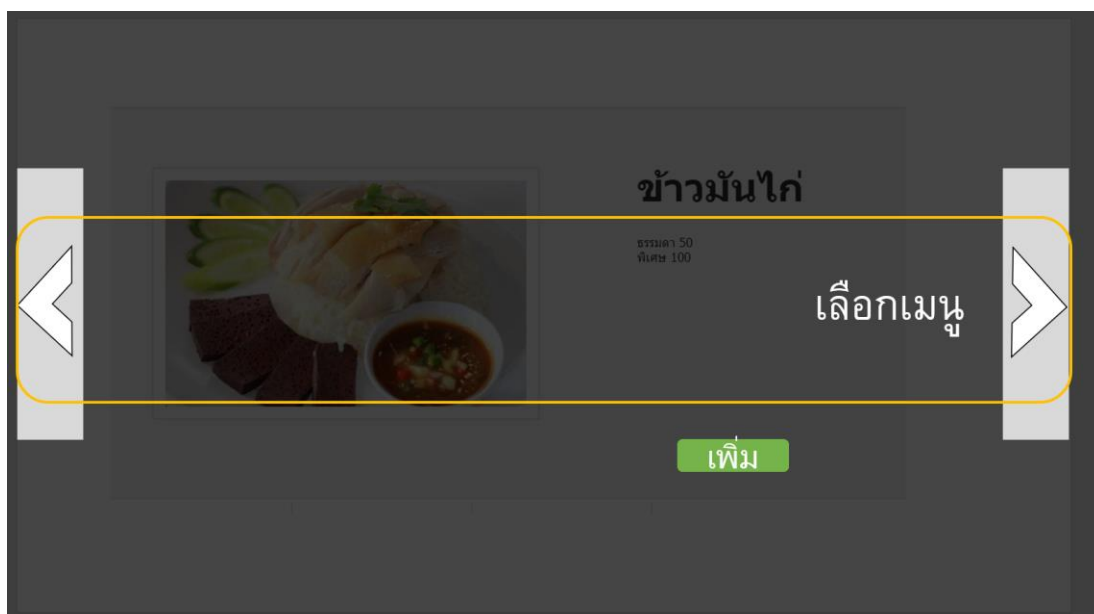
สร้างบัญชี

หน้าที่สอง เป็นหน้าเมนู มีรายละเอียดดังนี้

- มีการเลือกเมนูที่ต้องการ



- เมนูของแต่ละหมวดหมู่





ตะกร้าสินค้า

# Cart

### รายการอาหารของคุณ

ข้าวมันไก่	1	50
บาท		

### คำอธิบาย

🔥🔥 ขอเผ็ด ๆ 🔥🔥

รวม

บาท

50

สั่ง

ทางหลังร้านจะมีข้อมูลลูกค้าที่ลูกค้าได้สั่งออเดอร์ต่างๆเข้ามา

### รายการอาหาร

ข้าวมันไก่	1
------------	---

### คำอธิบาย

🔥🔥 ขอเผ็ด ๆ 🔥🔥

เสร็จ

# มีรายการอาหารที่

## ต้องทำสำหรับร้าน

กดปุ่มเมื่อทำครบ  
ทุกรายการ

- เมนูที่ทำเสร็จ

## หน้าเว็บของฝั่งลูกค้า



อาหารของคุณเสร็จแล้ว

ข้าวมัน

ราคา 50  
เมนู 100

มีข้อความเตือนเมื่อ  
ทางร้านทำครบทุก  
เมนูแล้ว

## Source Code

### FrontEnd (หลังร้าน)

#### 1.index.html

```
<!DOCTYPE
PE
html>

    <html lang="en">
        <head>
            <link
                rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css"
            />
            <link rel="stylesheet" href="/styleforindog.css" />

            <meta charset="UTF-8" />
            <meta http-equiv="X-UA-Compatible" content="IE=edge" />
            <meta name="viewport" content="width=device-width, initial-scale=1.0" />
            <title>Show Order</title>
        </head>
        <body>
            <!-- username -->
            <div class="table-users" style="font-family: 'Kanit', sans-serif">
                <div class="header">
                    <h4>
                        รายการอาหารที่สั่งของคุณ
                    <span style="color: rgb(132, 226, 255)" class="set-user"></span>
                    </h4>
                </div>

                <!-- ตาราง เมนู + จำนวน -->
                <table cellpadding="0">
                    <tr>
                        <td>รายการอาหาร</td>
                    </tr>
```

```

        <tr id="showTable">
            <td></td>
        </tr>

        <tr>
            <td>ราคา <span class="set-price"></span> บาท</td>
        </tr>
    </table>
</div>
<!-- คอมเม้น-ติดต่อ -->

    <div class="table-users" style="font-family: 'Kanit', sans-serif">
        <table cellpadding="0">
            <tr>
                <td class="set-comment"></td>
            </tr>
            <tr>
                <td class="set-contact"></td>
            </tr>
        </table>
    </div>

    <div>
        <!-- ปุ่มกดเพื่อไปออเดอร์อื่น -->
        <div class="frame">
            <button
                style="font-family: 'Kanit', sans-serif; width: 200px"
                class="custom-btn btn-11 click-next"
            >
                รายการถัดไป
            </button>
            <div class="dot"></div>
        </div>
        <center><h1 class="set-click"></h1></center>
    </div>
    <script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
    <script src="main.js"></script>
    </body>
</html>
<div class="ocean">

```

```

    <div class="wave"></div>
    <div class="wave"></div>
  </div>

```

## 2.main.js

```

let apiOrder =
"http://getapi-
project-2.app.ruk-
com.cloud/api/order";

let setUser = document.querySelector(".set-user"); // show
ww,as/ak
let setComment = document.querySelector(".set-comment");
// ขอเผื่อๆ
let setContact = document.querySelector(".set-contact");
// 09854022255
let setPrice = document.querySelector(".set-price"); //
09854022255
let setClick = document.querySelector(".set-click");
let clickNext = document.querySelector(".click-next");

getOurder();

async function getOurder() {
  let orderObj = await axios.get(apiOrder);
  ///////////////////////////////////////////////////
  let i = 0;
  //setUser.innerHTML = orderObj.data[i].user;
  function nextObj() {
    if (orderObj.data.length <= i) i = 0;
    setUser.innerHTML = orderObj.data[i].user;
    setClick.innerHTML = i + 1;
    show_menu(i); // table
    setPrice.innerHTML = orderObj.data[i].price;
    setComment.innerHTML = orderObj.data[i].comment;
    setContact.innerHTML = orderObj.data[i].contact;
    i++;
  }

  function show_menu(i) {
    let str = orderObj.data[i].orders;
    str = str.split("[").join("").split("]").join("");
  }
}

```

```

        str = str.split(",");
        showTable(str);
    }

    function unique(array) {
        return array.filter(function (el, index, arr) {
            return index == arr.indexOf(el);
        });
    }

    function showTable(arr) {
        document.getElementById("showTable").innerHTML = "";
        let count = {};
        arr.forEach(function (item) {
            item = item.split('').join("");
            count[item] = (count[item] || 0) + 1;
        });
        let unqStr = unique(arr);
        var body = document.getElementById("showTable");
        var tbl = document.createElement("table");
        var tblBody = document.createElement("tbody");
        unqStr.forEach((item, index) => {
            item = item.split('').join("");
            var row = document.createElement("tr");
            var cell = document.createElement("td");
            var cell2 = document.createElement("td");
            var cellText = document.createTextNode(item);
            var cellText2 =
document.createTextNode(count[item]);
            cell.appendChild(cellText);
            cell2.appendChild(cellText2);
            row.appendChild(cell);
            row.appendChild(cell2);
            tblBody.appendChild(row);
        });
        tbl.appendChild(tblBody);
        body.appendChild(tbl);
        tbl.setAttribute("border", "0");
    }

    clickNext.addEventListener("click", nextObj);
    //////////////////////////////////
    //set user

```

```
}
```

## Backend

### 1.index.js

```
var express =
require("express");

var cors = require("cors");
var mongoose = require("mongoose");
var bodyParser = require("body-parser");

var mongo_uri =

"mongodb+srv://root:12345@therest.ltsjd.mongodb.net/restaurant?retryWrites=true&w=majority";
mongoose.Promise = global.Promise;
mongoose.connect(mongo_uri, { useNewUrlParser: true }).then(
  () => {
    console.log("[success] task 2 : connected to the database ");
  },
  (error) => {
    console.log("[failed] task 2 " + error);
    process.exit();
  }
);

var app = express();

app.use(cors());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

var port = process.env.PORT || 5000;

app.listen(port, () => {
  console.log("[success] task 1 : listening on port " + port);
});

app.get("/", (req, res) => {
  res.status(200).send("หน้าแรกของ api express");
});
```

```

});

// path สำหรับ MongoDB ของเรา
var Order = require("./order_router");
app.use("/api/order", Order);

app.use((req, res, next) => {
  var err = new Error("ไม่พบ path ที่ต้องการ");
  err.status = 404;
  next(err);
});

```

## 2. order\_model.js

```

var mongoose =
require("mongoose");

var order_schema = mongoose.Schema(
{
  orders: {
    type: String,
  },

  user: {
    type: String,
  },
  comment: {
    type: String,
  },
  contact: {
    type: String,
  },

  price: {
    type: String,
  },
},
{
  collection: "orders",
}
);

```



```
var Order = mongoose.model("orders", order_schema);
module.exports = Order;
```

### 3. order\_router.js

```
var express =
require("express");

var router = express.Router();
var Order = require("./order_model");

// GET all
router.get("/", (req, res) => {
  Order.find().exec((err, data) => {
    if (err) return res.status(400).send(err);
    res.status(200).send(data);
  });
});

// GET 1
router.get("/:_id", (req, res) => {
  Order.findById(req.params._id).exec((err, data) => {
    if (err) return res.status(400).send(err);
    res.status(200).send(data);
  });
});

// POST (create new data)
router.post("/", (req, res) => {
  var obj = new Order(req.body);
  obj.save((err, data) => {
    if (err) return res.status(400).send(err);
    res.status(200).send("เพิ่มข้อมูลเรียบร้อยแล้ว");
  });
});

// PUT (update current data)
router.put("/:_id", (req, res) => {
  Order.findByIdAndUpdate(req.params._id, req.body, (err, data) => {
    if (err) return res.status(400).send(err);
    res.status(200).send("อัปเดตข้อมูลเรียบร้อยแล้ว");
  });
});
```

```

// DELETE (delete 1 data)
router.delete("/:id", (req, res) => {
  Order.findByIdAndDelete(req.params._id, (err, data) => {
    if (err) return res.status(400).send(err);
    res.status(200).send("ลบข้อมูลเรียบร้อยแล้ว");
  });
});

module.exports = router;

```

## FrontEnd และ BackEnd (หน้าร้าน มันรวมกันอยู่)

### 1. login.html

```

<!DOCTYPE
PE
html>

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>

    <style>
      @import
url("https://fonts.googleapis.com/css?family=Muli&display=swap")
;
      @import
url("https://fonts.googleapis.com/css?family=Open+Sans:400,500&d
isplay=swap");

      * {
        box-sizing: border-box;
      }

      body {
        background-color: #59b65e;
        font-family: "Open Sans", sans-serif;

```

```
display: flex;
align-items: center;
justify-content: center;
min-height: 100vh;
margin: 0;
}

.container {
background-color: #fff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3);
overflow: hidden;
width: 400px;
max-width: 100%;
}

.header {
border-bottom: 1px solid #f0f0f0;
background-color: #f7f7f7;
padding: 20px 40px;
}

.header h2 {
margin: 0;
}

.form {
padding: 30px 40px;
}

.form-control {
margin-bottom: 10px;
padding-bottom: 20px;
position: relative;
}

.form-control label {
display: inline-block;
margin-bottom: 5px;
}

.form-control input {
border: 2px solid #f0f0f0;
```

```

        border-radius: 4px;
        display: block;
        font-family: inherit;
        font-size: 14px;
        padding: 10px;
        width: 100%;
    }

    .form-control input:focus {
        outline: 0;
        border-color: #777;
    }

    .form-control i {
        visibility: hidden;
        position: absolute;
        top: 40px;
        right: 10px;
    }

    .form-control.success i.fa-check-circle {
        color: #2ecc71;
        visibility: visible;
    }

    .form button {
        text-align: center;
        background-color: #59b65e;
        border: 2px solid #59b65e;
        border-radius: 4px;
        color: #fff;
        display: block;
        font-family: inherit;
        font-size: 16px;
        padding: 10px;
        margin-top: 20px;
        width: 100%;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="header">

```

```

        <h2>Log In</h2>
    </div>
    <form id="form" class="form" method="POST" action="/">
        <div class="form-control">
            <label for="username">Username</label>
            <input type="text" name="username"
placeholder="scooter_xd" id="id" />
        </div>

        <div class="form-control">
            <label for="username">Password</label>
            <input
                type="password"
                placeholder="12345678"
                id="password"
                name="password"
            />
        </div>

        <button name="submit" value="yes">Submit</button>
        <input
            id="log"
            type="submit"
            name="regis"
            style="border: 0; background-color: white"
            value="Don't have Account"
        />

        <!-- <input type="submit" value="Submit" /> -->
    </form>
</div>
<script>
    if (document.getElementById("id").value == "log")
        console.log("xxxxxxxxxxxxxxxxxxxx");
        console.log(document.getElementById("id").value + " <<<<
user name");
    </script>
</body>
</html>

```

## 2.index.js

```
require("dotenv").config();

const express = require("express");
const app = express();
const port = process.env.PORT;

app.get("/", (req, res) => {
  res.send("Hello node.js  
World!ssssssssssssssssssss");
});

app.listen(port, () => {
  console.log(`Example app listening  
http://localhost:${port}`);
});
```

## 3.index.ejs

```
<!DOCTYPE
html>

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>sale food ja</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Kanit:wght@300&dis
play=swap" rel="stylesheet">

    <!-- test -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/moder
nizr.min.js" type="text/javascript"></script>

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/norm
alize.min.css">
    <link rel="stylesheet" href="/assets/style.css">
    <!-- test -->
```

```

<!-- Import Styles -->
<link rel="stylesheet" href="/assets/styles.css" />
<!-- Import Vue.js -->
<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css"
  rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+sc0/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
  crossorigin="anonymous"
/>
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7Z0dauHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"
></script>
<script src="https://unpkg.com/vue@3.0.0-
beta.12/dist/vue.global.js"></script>
</head>
<body style="background: radial-gradient(ellipse at left,
#181b20 0%, #090a0f 10%, #34344d 50%,#3d3b75 80%,#34344d 100%
)">

  <!-- test -->
  <link
href='https://fonts.googleapis.com/css?family=Lato:300,400,700'
rel='stylesheet' type='text/css'>
    <div id='stars'></div>
    <div id='stars2'></div>
    <div id='stars3'></div>

    <script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.m
in.js'></script><script src='./script.js'></script>
    <!-- test -->

<nav>
  <br />

```

```

        <h1 style="float: right; margin-right:
10%;color:white;font-family: 'Kanit', sans-serif;">สวัสดิ์คุณ <%= id
%></h1>
        <br/>
        <div class="frame">
            <form action="/" method="POST">
                <input type="hidden" name="logout" value="logout">
                <button style=" margin-right: 7%;"
                class="custom-btn btn-11">
                    logout
                <div class="dot"></div></button>
            </form>

        </div>
    </nav>
    <div id="app">
        <center>

    </nav>
    <div id="app">
        <center>
            
        </center>
        <center>
            <div class="contente" style="font-family: 'Kanit', sans-
serif;">
                <div class="slider1-wrapper" >
                    ร้านเรามี
                <div class="slider1">
                    <div class="slider1-text11">ข้าว</div>
                    <div class="slider1-text21">ถ้วยเตี๋ย</div>
                    <div class="slider1-text31">หอย</div>

                </center>

            <bubletea
style="font-family: 'Kanit', sans-serif;"
:stocker="stocker"
@add-to-cart="updateCart"
@remove-from-cart="removeById"
@add-to-priced="updatePriced"
@remove-from-priced="removeByIdp"

```



```

>
  <!-- solution -->
</bubletea>

<rice
style="font-family: 'Kanit', sans-serif;"
  :stocker="stocker"
  @add-to-cart="updateCart"
  @remove-from-cart="removeById"
  @add-to-priced="updatePriced"
  @remove-from-priced="removeByIdp"
>
  <!-- solution -->
</rice>

<noodle
style="font-family: 'Kanit', sans-serif;"
  :stocker="stocker"
  @add-to-cart="updateCart"
  @remove-from-cart="removeById"
  @add-to-priced="updatePriced"
  @remove-from-priced="removeByIdp"
>
  <!-- solution -->
</noodle>

<center>
  <div class="cart2" style="width: 700px;font-family:
'Kanit', sans-serif;">
    <center>
      <h2>รายการอาหารที่สั่ง</h2>
    </center>

    <div style="display: none">
      {{sum = 0}}
    </div>

    <div v-for="(item, index) in cart">
      <h4>
        <div style="float: left">
          {{ item }}
        </div>
        <div style="float: right">

```

```

        {{ priced[index] }} บาท
    </div>
</h4>
<br />
<div style="display: none">
    {{ sum += priced[index] }}
</div>
</div>
<br />
<h4>
    <div style="float: left">รวม: </div>
    <div style="float: right"> {{ sum }} บาท</div>
</h4>

</div>

<!-- button -->
<form action="/" method="post">

    <!-- {{orderStr = cart + "" }} -->
    <input type="hidden" name="order" :value="toStr" />
    <input type="hidden" name="orderTotal" :value="getsum"
/>

    <input type="hidden" name="orderByUser" value="<%= id
%>" />

    <div style="width: 400px">
        <h3>เบอร์โทร</h3>
        <input type="tel" name="orderContact" />

        <h3>รายละเอียดเพิ่มเติม</h3>

        <input type="text" name="orderComment" />
    </div>
    <br />
    <button
        type="submit"
        name="addToDB"
        value="addToDB"
        class="go"
        class="custom-btn btn-11"
        :disabled="cart.length > 0 ?disabled :'' "
    >
        Success

```

```

        </button>

    </form>
</center>
<br/>
<br/>
<!-- Import App -->
<script src="/main.js"></script>

<!-- Import Components -->
<script src="/components/water.js"></script>
<script src="/components/tess.js"></script>
<script src="/components/noodle.js"></script>
<script src="/components/rice.js"></script>
<script src="/components/button.js"></script>

<!-- Mount App -->
<script>
    const mountedApp = app.mount("#app");
</script>
</div>
</body>

<br/>
</html>

```

#### 4.sever.js

```

const
express =
require("exp
ress");

const app = express();
const ejs = require("ejs");
const path = require("path");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
app.set("view engine", "html");
app.engine("html", require("ejs").renderFile);
app.use(express.static(path.join(__dirname, "public")));

```

```

app.use(bodyParser.urlencoded({ extended: true }));

mongoose.connect(

  "mongodb+srv://root:12345@therest.ltsjd.mongodb.net/restaura
nt?retryWrites=true&w=majority",
  { useNewUrlParser: true },
  { useUnifiedTopology: true }
);

// schema for member
const memberSchema = {
  username: String,
  password: String,
};
const orderSchema = {
  orders: String,
  user: String,
  comment: String,
  contact: String,
  price: String,
};

const Member = mongoose.model("Member", memberSchema);
const Order = mongoose.model("Order", orderSchema);

let url = "/login.html";
let id = "";
let checker = ""; // for check username in mongodb - for
register

app.get("/", function (req, res) {
  // console.log("xxxxxxxxxxxxxxxxxxxxxx" + req.body.regis +
  "ss");

  if (req.body.regis) {
    url = "/regis.ejs";
    console.log("regis");
    // res.redirect("/regis.html");
    res.render(__dirname + url, { checker: checker });
  } else {
    res.render(__dirname + url, { id: id, checker: checker
  }); //login

```

```

    }
  });

  //post
  app.post("/", function (req, res) {
    if (req.body.regis) {
      url = "/regis.ejs";
    }

    // จากลือกอินฟอร์ม
    ///////////////////////////////////
    //for register
    if (url == "/regis.ejs") {
      if (req.body.username + "").trim() === "") {
        //console.log("wrong");
        res.redirect("/");
      } else if (req.body.password + "").trim() === "") {
        res.redirect("/");
        //console.log("wrong");
      } else if (req.body.password !== req.body.password2) {
        res.redirect("/");
        //console.log("wrong");
      } else {
        run();
        checker = "";
        async function run() {
          await Member.find(async function (err, data) {
            for await (const key of data) {
              // console.log(key.username);
              if (req.body.username === key.username) {
                console.log(key.username + " <<");
                checker = "fail";
                console.log(checker + " 1");
                // console.log(url);
                break;
              }
            }
          });
        }
      }
    }
  });

  console.log(" 2 jaaaa");

  console.log(checker);
  if (checker !== "fail") {

```

```

        console.log(checker + " 3 not fail");
        url = "/login.html";
        console.log("save");
        let NewMem = new Member({
            username: req.body.username,
            password: req.body.password,
        });
        NewMem.save();
    }
    console.log(checker + " <<");
    res.redirect("/");
}
}
} else {
    //if (url == "/login.html")
    ////

    if (req.body.logout) {
        console.log("logout __>" + req.body.logout);
        console.log("reset");
        let url = "/login.html";
        let id = "";
        let checker = "";
        res.render(__dirname + url, { id: id, checker: checker
    });

    ///
} else if (req.body.addToDB) {
    // console.log(req.body.addToDB);
    console.log("can");
    console.log(req.body.order);
    console.log(req.body.orderByUser);
    console.log(req.body.orderTotal);
    url = "/public/index.ejs";
    res.redirect("/");
    let Neworder = new Order({
        orders: req.body.order,
        user: req.body.orderByUser,
        comment: req.body.orderComment,
        contact: req.body.orderContact,
        price: req.body.orderTotal,
    });
    Neworder.save();

```

```

    } else {
        /////////////// for login
        //ตรวจรหัส
        //เปลี่ยนไปหน้าร้านค้า
        console.log("login");
        Member.find(function (err, data) {
            // if (err) return console.error(err);
            for (const key of data) {
                if (req.body.username === key.username)
                    if (req.body.password === key.password) {
                        // console.log("ww");
                        id = req.body.username;
                        // console.log(id + " << id");
                        url = "/public/index.ejs";

                        break;
                    } else {
                        console.log("password wrong jaa");
                        break;
                    }
                else {
                }
            }
            res.redirect("/");
        });
    }
});

app.listen(3000, function () {
    console.log("server running on port 3000");
});

```

## 5. regis.ejs

```
<!DOCTYPE
PE
html>

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />

    <style>
      @import
url("https://fonts.googleapis.com/css?family=Muli&display=swap")
;
      @import
url("https://fonts.googleapis.com/css?family=Open+Sans:400,500&d
isplay=swap");
      * {
        box-sizing: border-box;
      }
      body {
        background-color: #9b59b6;
        font-family: "Open Sans", sans-serif;
        display: flex;
        align-items: center;
        justify-content: center;
        min-height: 100vh;
        margin: 0;
      }
      .container {
        background-color: #fff;
        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3);
        overflow: hidden;
        width: 400px;
        max-width: 100%;
```



```
}
.header {
  border-bottom: 1px solid #f0f0f0;
  background-color: #f7f7f7;
  padding: 20px 40px;
}
.header h2 {
  margin: 0;
}
.form {
  padding: 30px 40px;
}
.form-control {
  margin-bottom: 10px;
  padding-bottom: 20px;
  position: relative;
}
.form-control label {
  display: inline-block;
  margin-bottom: 5px;
}
.form-control input {
  border: 2px solid #f0f0f0;
  border-radius: 4px;
  display: block;
  font-family: inherit;
  font-size: 14px;
  padding: 10px;
  width: 100%;
}
.form-control input:focus {
  outline: 0;
  border-color: #777;
}
.form-control.success input {
  border-color: #2ecc71;
}
.form-control.error input {
  border-color: #e74c3c;
}
.form-control i {
  visibility: hidden;
  position: absolute;
```

```

        top: 40px;
        right: 10px;
    }
    .form-control.success i.fa-check-circle {
        color: #2ecc71;
        visibility: visible;
    }
    .form-control.error i.fa-exclamation-circle {
        color: #e74c3c;
        visibility: visible;
    }
    .form-control small {
        color: #e74c3c;
        position: absolute;
        bottom: 0;
        left: 0;
        visibility: hidden;
    }
    .form-control.error small {
        visibility: visible;
    }
    .form button {
        text-align: center;
        background-color: #8e44ad;
        border: 2px solid #8e44ad;
        border-radius: 4px;
        color: #fff;
        display: block;
        font-family: inherit;
        font-size: 16px;
        padding: 10px;
        margin-top: 20px;
        width: 100%;
    }
</style>

<title>สมัคร</title>
</head>
<body>
    <input type="hidden" id="checker" value="<%= checker %>" />
    <div class="container">
        <div class="header">
            <h2>Create Account</h2>

```

```

</div>
<form
  id="form"
  class="form"
  onsubmit="return validateMyForm();"
  method="POST"
  action="/"
>
  <div class="form-control">
    <label for="username">Username</label>
    <input
      type="text"
      name="username"
      placeholder="scooter_xd"
      id="username"
    />
  </div>

  <div class="form-control">
    <label for="password">Password</label>
    <input
      type="password"
      placeholder="12345678"
      id="password"
      name="password"
    />
  </div>

  <div class="form-control">
    <label for="password">Confirm Password</label>
    <input
      type="password"
      placeholder="12345678"
      name="password2"
      id="password2"
    />
  </div>

  <button name="submit">Submit</button>
</form>
<script type="text/javascript">
  let checker = document.getElementById("checker").value;
  console.log("checker > " + checker);
  if (checker == "fail") {
    console.log(checker + " < say");
  }
</script>

```

```

        alert("fail this username has already been taken");
    }
    function validateMyForm() {
        if (document.getElementById("username").value +
        "".trim() === "")
            return setErr("username is blank");
        if (document.getElementById("password").value +
        "".trim() === "")
            return setErr("password is blank");
        else if (
            document.getElementById("password").value +
        "".trim() !==
            document.getElementById("password2").value +
        "".trim()
        )
            return setErr("password is not match");
        //location.href = "login.htm";
        return true;
    }
    function setErr(mess) {
        alert(mess);
        return false;
    }
}
</script>
</div>
</body>
</html>

```

## 6.main.js

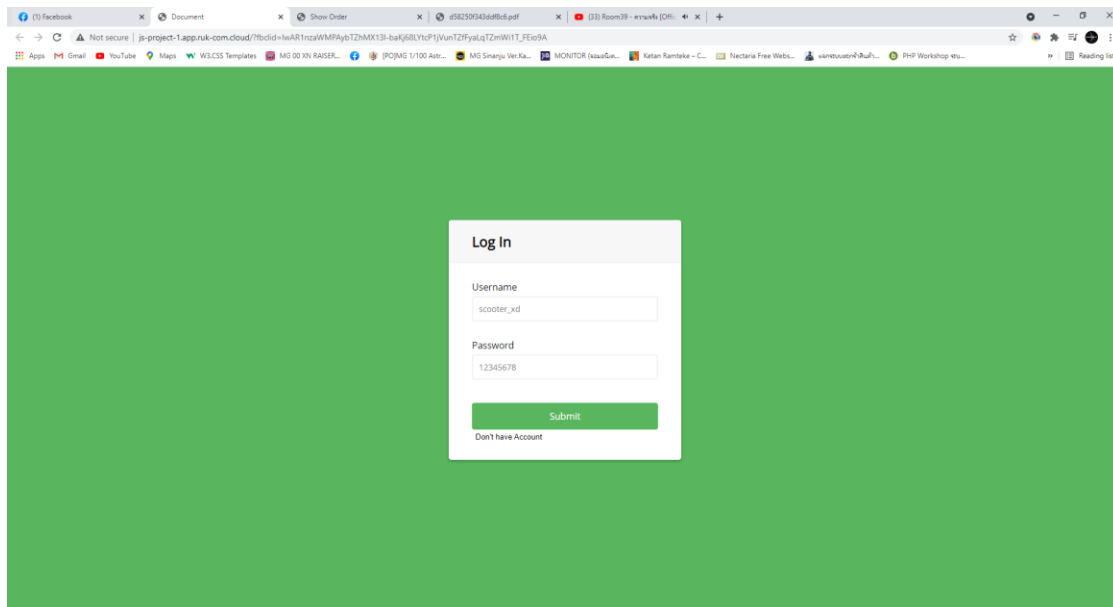
```
const app =
Vue.createApp({
  data() {
    return {
      sum: 0,
      priced: [],
      cart: [],
      stocker: true,
      orderStr: "",
    };
  },
  methods: {
    updateCart(id) {
      this.cart.push(id);
    },
    updatePriced(price) {
      this.priced.push(price);
    },
    // solution
    removeById(id) {
      const index = this.cart.indexOf(id);
      if (index > -1) {
        this.cart.splice(index, 1);
      }
    },
    // solution
  },
  computed: {
    toStr() {
      return (this.orderStr = JSON.stringify(this.cart));
    },
    getsum() {
      return this.sum;
    },
  },
},
```

```
});
```

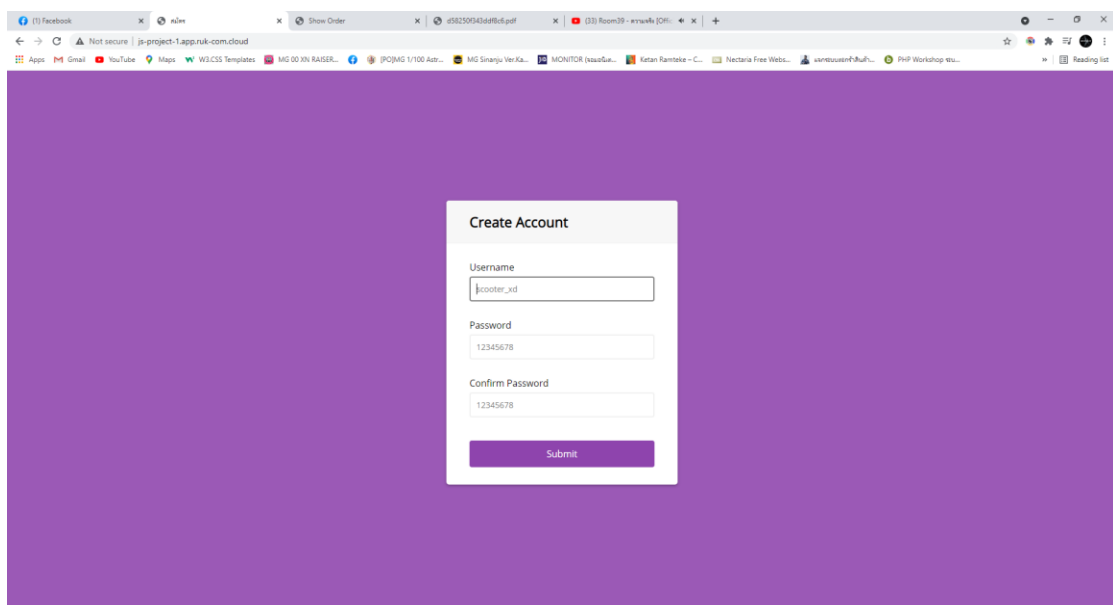
## แสดงตัวอย่างหน้าเว็บ

### หน้าร้าน

หน้า log-in เป็นหน้าให้ลูกค้าได้ log-in เพื่อเข้าสู่ระบบของเว็บเพื่อสั่งอาหารแต่ถ้าไม่มีข้อมูลลูกค้าใน database จะทำการเปลี่ยนเป็นหน้าให้สมัครสมาชิกแทนและเมื่อสมัครเสร็จก็จะกลับมายังหน้า log-in

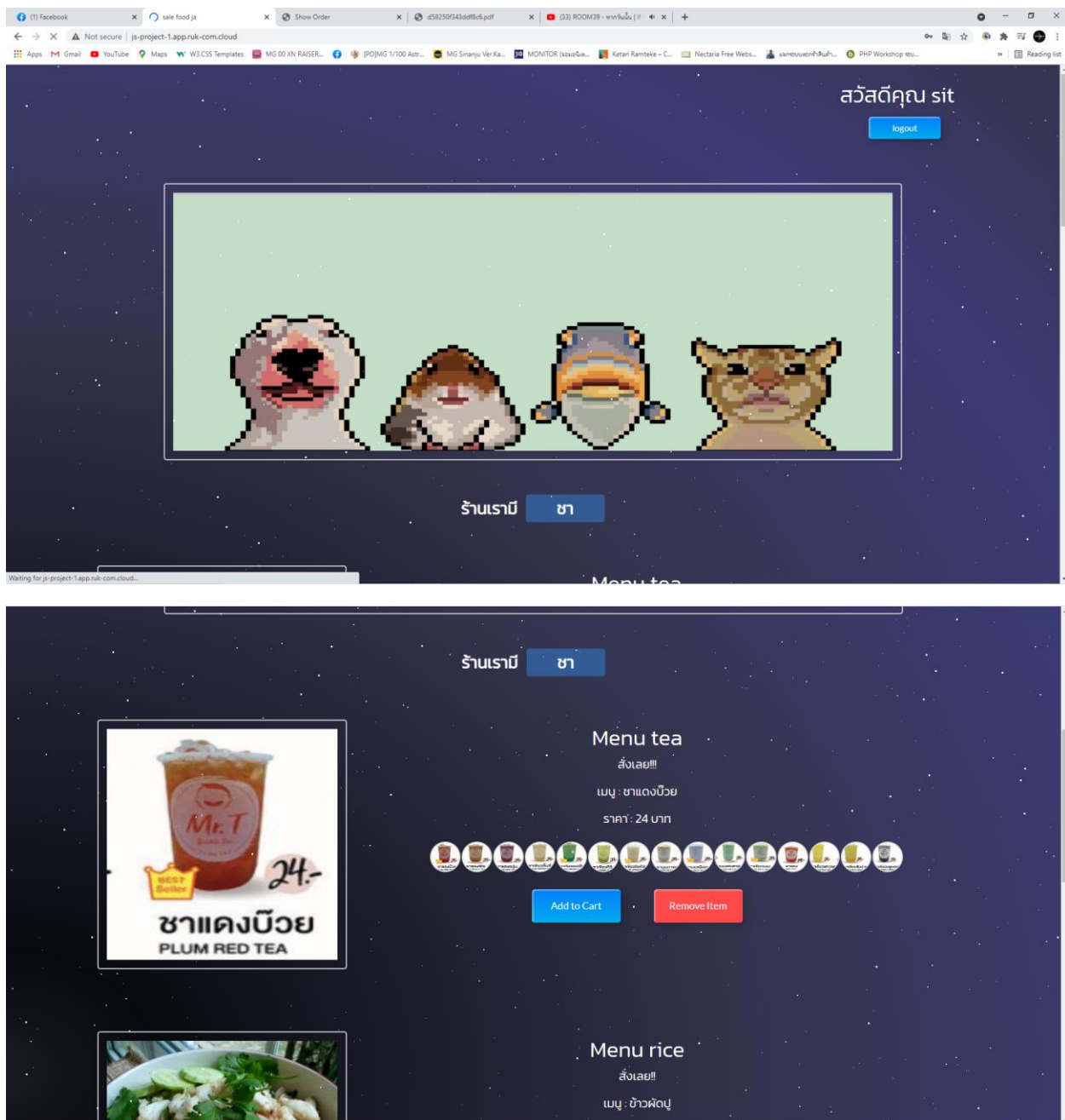


### หน้า sign-in

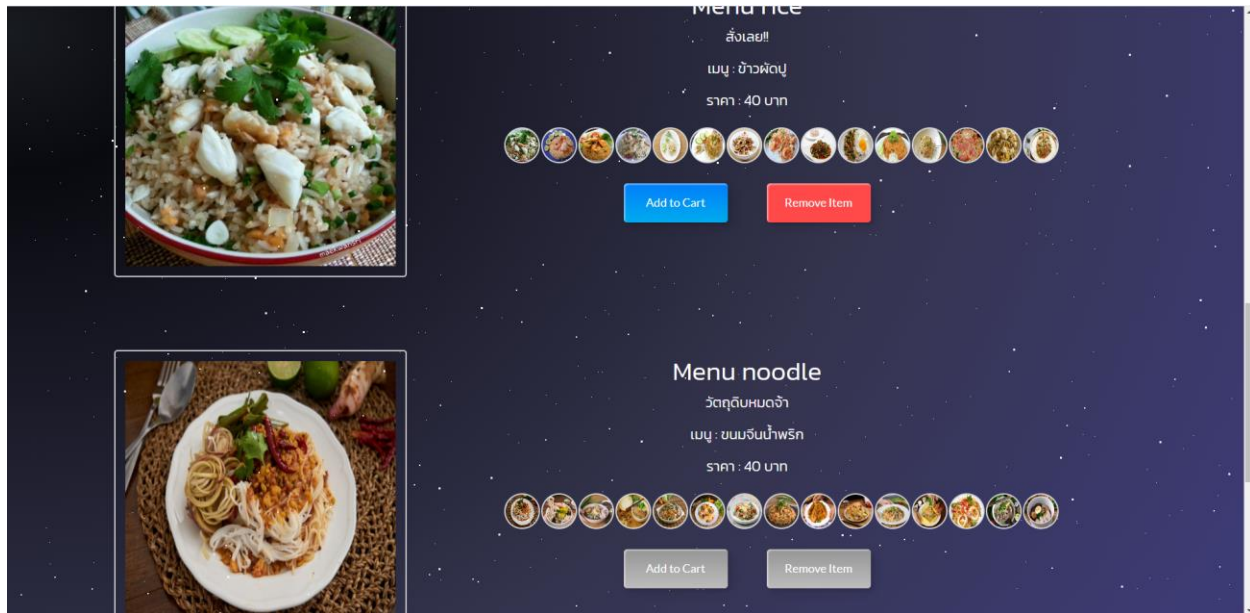


## หน้าสินค้า

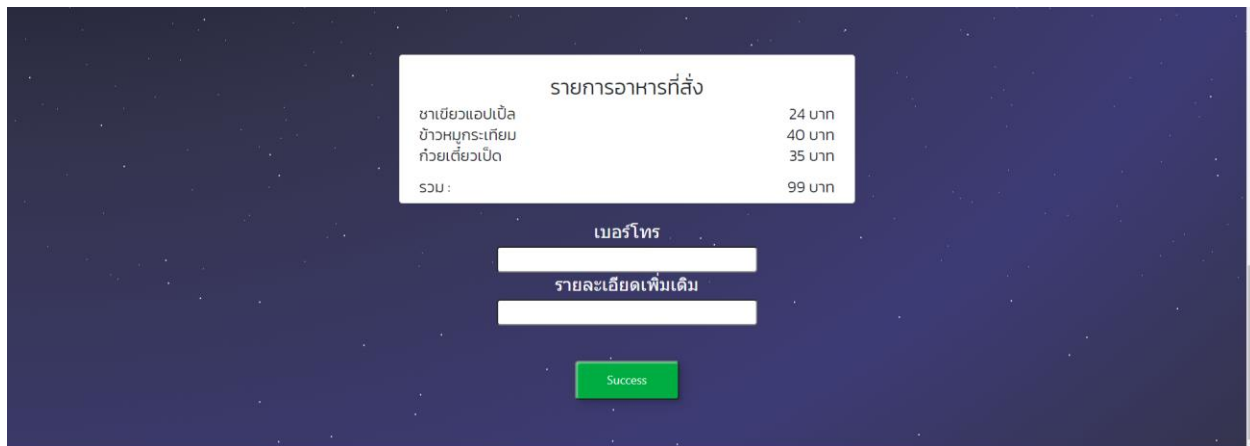
หน้าสินค้าสามารถเลือกเมนูอาหารต่างๆไม่ว่าจะเป็นเมนูเส้นเมนูข้าวหรือเมนูน้ำและเมื่อเราเลือกเมนูที่ต้องการแล้วสามารถกดปุ่ม Add To Cart ได้เลยแต่ถ้าเมนูไหนที่ไม่มีของในสต็อกก็ไม่สามารถกด Add To Cart ได้และเมื่อกดเสร็จข้อมูลต่างๆจะถูกส่งไปยังลิสอเดอร์ด้านล่าง





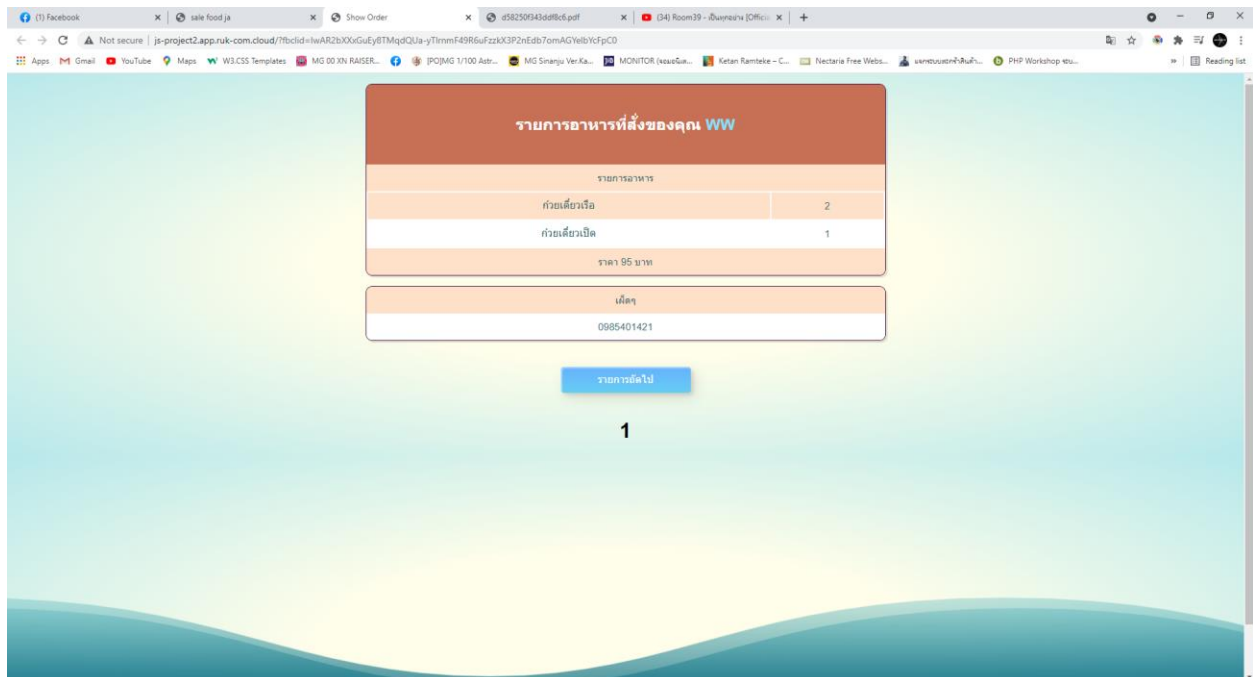


ออเดอร์ที่สั่งพร้อมกรอกข้อมูลเกี่ยวกับเบอร์โทร และ รายละเอียดเพิ่มเติม และเมื่อกดสั่งจะมีปุ่ม success และข้อมูลต่างๆจะส่งขึ้น database และส่งข้อมูลออเดอร์ไปยังหลังร้าน



## หลังร้าน

การดูออเดอร์ของลูกค้าที่สั่งเข้ามาสามารถดูออเดอร์ถัดไปเรื่อยๆทุกเมนูที่ลูกค้าสั่งและสามารถนำ contact ของลูกค้าและรายละเอียดต่างๆมาดูได้และเมื่อทำเสร็จก็ให้แม่ค้าโทรติดต่อหาลูกค้าได้



## บรรณานุกรม

- 1) mongoDB คืออะไร ?

<https://www.techtalkthai.com/mongodb-announced-atlas-mongodb-as-a-service/>

- 2) Vue.js คืออะไร ?

<https://devahoy.com/blog/2019/08/introduction-to-vuejs/>

- 3) Node.js คืออะไร ?

<https://devahoy.com/blog/2015/07/getting-started-with-nodejs/>

- 4) Postman คืออะไร ?

<https://medium.com/@nest12345/%E0%B8%A1%E0%B8%B2%E0%B8%97%E0%B8%94%E0%B8%AA%E0%B8%AD%E0%B8%9A-api-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-postman-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-c750a09c86ce>

- 5) Vs Code คืออะไร ?

<http://cs.bru.ac.th/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89-visual-studio-code-2/>

- 6) GitHub คืออะไร ?

<https://saixiii.com/what-is-github/>