

ฐานข้อมูลบริษัทน้ำเปล่า

นาย มนูญศ จิตรสมาน 116510901011-6

นางสาว อจิรวดี จันทวรรณ 116510901026-4

นาย วรรณนที แดงรัมย์ 116510901027-2

เสนอ

ดร.รัฐพรหม พรหมคำ

สาขาวิชาคณิตศาสตร์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

November 2, 2024

สารบัญ

1	ฐานข้อมูลบริษัทผลิตน้ำเปล่า	2
1.1	ปัญหา	2
1.2	ER Diagram	3
2	SQL	4
2.1	SQL ที่ใช้ในการแก้ปัญหา	5
3	Data On Website	7
3.1	Code ในการนำขึ้นเว็บไซต์	7
4	Data On GIT	18

บทที่ 1

ฐานข้อมูลบริษัทผลิตน้ำเปล่า

Database หรือ ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆ แฟ้มข้อมูล ซึ่งถูกจัดเก็บอย่างเป็นระบบ โดยมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล ทำให้ผู้ใช้สามารถใช้ข้อมูลได้อย่างมีประสิทธิภาพ

ทำไมต้องเป็นบริษัทผลิตน้ำเปล่า เนื่องจากน้ำเป็นสิ่งที่ขาดไม่ได้สำหรับมนุษย์ การมีบริษัทซึ่งคอยผลิตและส่งออกน้ำจึงเป็นเรื่องที่มีอยู่ทั่วไป ไม่ว่าจะเป็นร้านค้าขนาดเล็กจนไปถึงบริษัทขนาดใหญ่ การมีฐานข้อมูลสำหรับตรวจสอบการผลิตและส่งออกน้ำจึงเป็นเรื่องปกติ

ฐานข้อมูลบริษัทผลิตน้ำเปล่านี้ มีข้อมูลของทั้ง ลูกค้า, สินค้า, ยอดขาย, การชำระเงิน, คำสั่งซื้อ, ขนส่ง และ เส้นทางของสินค้า ซึ่งเป็นฐานข้อมูลที่มีข้อมูลแบบที่ต้องการ เพื่อแก้ไขปัญหา และความสะดวกของผู้ใช้งาน

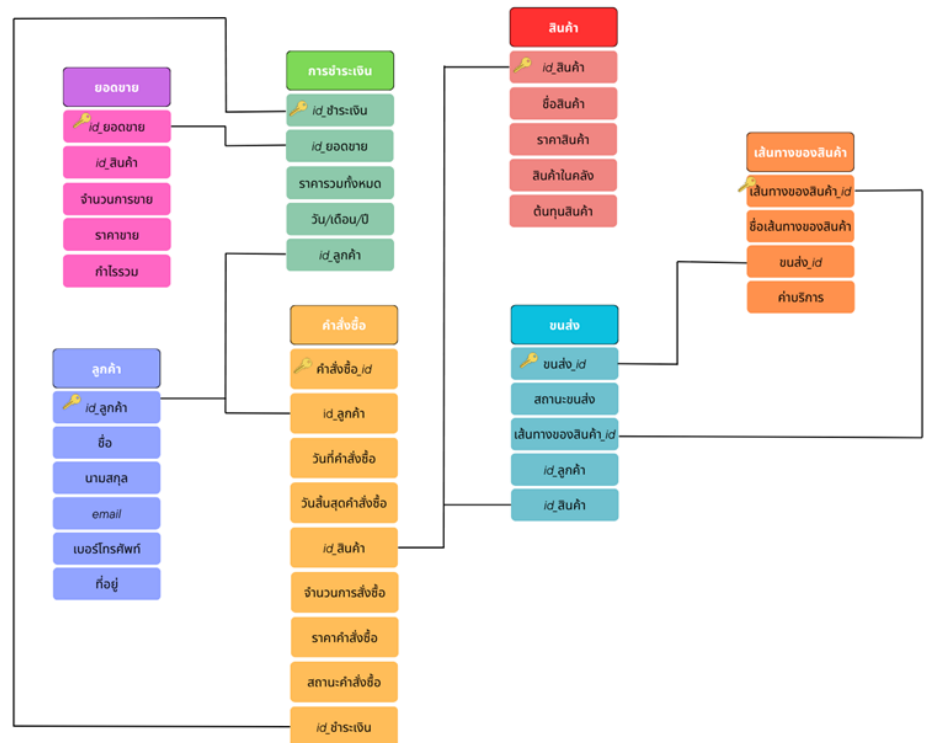
1.1 ปัญหา

ปัญหาที่ถูกตั้งขึ้น

1. ลูกค้าจะสั่งสินค้าในช่วงเดือนไหนมากที่สุด
2. ถ้าไรที่ได้แต่ละเดือนมีแนวโน้มเพิ่มขึ้นหรือลดลง
3. การขนส่งเดือนไหนที่ค่าบริการมากที่สุด

4. ลูกค้าคนไหนที่สั่งซื้อมากที่สุดในแต่ละเดือน

1.2 ER Diagram



รูปภาพ 1.1: ER Diagram

บทที่ 2

SQL

SQL เป็นภาษาโปรแกรมสำหรับจัดเก็บและประมวลผลข้อมูลในฐานข้อมูลแบบเชิงสัมพันธ์ ฐานข้อมูลแบบเชิงสัมพันธ์เก็บข้อมูลในรูปแบบตารางที่มีแถวและคอลัมน์ที่เป็นตัวแทนของหมวดข้อมูลที่แตกต่างกันและความสัมพันธ์ต่างๆ ระหว่างค่าข้อมูล สามารถใช้คำสั่ง SQL ในการจัดเก็บ ปรับปรุง ลบ ค้นหา และดึงข้อมูลจากฐานข้อมูล นอกจากนี้ยังสามารถใช้ SQL ในการรักษาและเพิ่มประสิทธิภาพการทำงานของฐานข้อมูล

ตัวอย่างในการสร้าง SQL ที่ใช้ในการสร้างฐานข้อมูลบริษัทผลิตน้ำเปล่า ในที่นี้จะใช้เป็นตารางของลูกค้ามายกตัวอย่าง

```
CREATE TABLE Customers (  
    Customer_id INT PRIMARY KEY,  
    Firstname TEXT,  
    Lastname TEXT,  
    Email TEXT,  
    Phone_number NUMERIC,  
    Address TEXT  
);
```

และตัวอย่างในการใส่ข้อมูลลงในตาราง จากตัวอย่างด้านบน

```
INSERT INTO users (Customer_id, Firstname, Lastname, Email, Phone_number, Address) VALUES
(1, 'Somchai', 'Sukjai', 'somchai@example.com', '812345678', 'Bangkok, Thailand'),
(2, 'Sompong', 'Dee', 'sompong@example.com', '823456789', 'Chiang Mai, Thailand'),
(3, 'Malee', 'Sri', 'malee@example.com', '834567890', 'Phuket, Thailand'),
-- ... (ข้อมูลที่เหลือ 4-30)
(30, 'Chalerm', 'Fon', 'chalerm@example.com', '801234569', 'Trat, Thailand');
```

2.1 SQL ที่ใช้ในการแก้ปัญหา

1. ลูกค้าจะสั่งซื้อสินค้าในช่วงเดือนไหนมากที่สุด

The screenshot shows a SQL query in the SQL Editor window:

```
1 SELECT
2   Order_Month,
3   count(*) AS total_orders
4 FROM
5   Orders
6 GROUP BY
7   Order_Month
8 ORDER BY
9   total_orders DESC;
```

The Results window displays the following data:

Order_Month	total_orders
สิงหาคม	5
ธันวาคม	2
กุมภาพันธ์	2
พฤษภาคม	2
ตุลาคม	2

The DB Schema window on the right shows the database structure:

Name	Type	Schema
Customers	CREATE TABLE	Cust
Orders	CREATE TABLE	Ord
Payment	CREATE TABLE	Pay
Product	CREATE TABLE	Prod
Sales	CREATE TABLE	Sale
Supply_chain	CREATE TABLE	Sup
Transport	CREATE TABLE	Tran

2. กำไรที่ได้แต่ละเดือนมีแนวโน้มเพิ่มขึ้นหรือลดลง

The screenshot shows a SQL query in the SQL Editor window:

```
1 SELECT
2   Order_Month,
3   SUM (Order_Price - Product_Cost * Order_Count) AS Total_Profit
4 FROM
5   Orders
6 JOIN
7   Product ON Orders.Product_Id = Product.Product_Id
8 GROUP BY
9   Order_Month
10 ORDER BY
11   Order_Month;
```

The Results window displays the following data:

Order_Month	Total_Profit
กุมภาพันธ์	-1445
พฤษภาคม	154
ธันวาคม	104
สิงหาคม	-50
พฤษภาคม	-2576

The DB Schema window on the right shows the database structure:

Name	Type	Schema
Customers	CREATE TABLE	Cust
Orders	CREATE TABLE	Ord
Payment	CREATE TABLE	Pay
Product	CREATE TABLE	Prod
Sales	CREATE TABLE	Sale
Supply_chain	CREATE TABLE	Sup
Transport	CREATE TABLE	Tran

3. การขนส่งเดือนไหนที่ค่าบริการมากที่สุด

The screenshot shows a SQL query in the 'Query' tab of SQL Server Enterprise Manager. The query is a SELECT statement that joins the Orders, Transport, and Supply_chain tables to calculate the total service fee by month. The results are displayed in a table with 5 rows, showing the month and the total service fee.

```

1 SELECT
2     Order_Month,
3     SUM (Service_fee) AS total_service_fee
4 FROM Orders
5 JOIN Transport
6 ON Orders.Product_Id = Transport.Product_Id
7 JOIN Supply_chain
8 ON Transport.Transport_Id = Supply_chain.Transport_Id
9 GROUP BY Order_Month
10 ORDER BY total_service_fee DESC;

```

Order_Month	total_service_fee
1 สิงหาคม	6165
2 มกราคม	2310
3 พฤศจิกายน	1425
4 เมษายน	1360
5 ธันวาคม	1290

Execution finished without errors.
Result: 5 rows returned in 12ms
At line 1:

```

SELECT
    Order_Month,
    SUM (Service_fee) AS total_service_fee
FROM Orders
JOIN Transport
ON Orders.Product_Id = Transport.Product_Id
JOIN Supply_chain
ON Transport.Transport_Id = Supply_chain.Transport_Id
GROUP BY Order_Month
ORDER BY total_service_fee DESC;

```

The right-hand pane shows the 'DB Schema' for the 'MAIL' database, listing tables such as Customers, Orders, Payment, Product, Sales, Supply_chain, and Transport.

4. ลูกค้าคนไหนที่สั่งซื้อมากที่สุดในแต่ละเดือน

The screenshot shows a SQL query in the 'Query' tab of SQL Server Enterprise Manager. The query is a SELECT statement that joins the Customers and Orders tables to find the top customers by order price for each month. The results are displayed in a table with 5 rows, showing the customer ID, full name, order month, and order price.

```

1 SELECT
2     A.Customer_Id,
3     A.FirstName || ' ' || A.LastName AS FullName,
4     B.Order_Month,
5     B.Order_Price
6 FROM Customers AS A
7 JOIN Orders AS B
8 ON A.Customer_Id = B.Customer_Id
9 ORDER BY Order_Price DESC;

```

Customer_Id	FullName	Order_Month	Order_Price
8 Suda Lee	สอดา ลี	มกราคม	425
4 Wichai Manee	วิชัย มณี	พฤศจิกายน	410
4 Wichai Manee	วิชัย มณี	ธันวาคม	375
8 Suda Lee	สอดา ลี	สิงหาคม	355
9 Nid Ket	นิด เกต	มกราคม	330

Execution finished without errors.
Result: 10 rows returned in 8ms
At line 1:

```

SELECT
    A.Customer_Id,
    A.FirstName || ' ' || A.LastName AS FullName,
    B.Order_Month,
    B.Order_Price
FROM Customers AS A
JOIN Orders AS B
ON A.Customer_Id = B.Customer_Id
ORDER BY Order_Price DESC;

```

The right-hand pane shows the 'DB Schema' for the 'MAIL' database, listing tables such as Customers, Orders, Payment, Product, Sales, Supply_chain, and Transport.

บทที่ 3

Data On Website

การนำฐานข้อมูลขึ้นบนเว็บไซต์ คือ กระบวนการเชื่อมต่อฐานข้อมูลที่มีอยู่ภายในเครื่อง, เซิร์ฟเวอร์ หรือ ที่จัดเก็บข้อมูลภายนอก เช่น แฟลชไดรฟ์ และนำเสนอในรูปแบบเว็บไซต์ ทำให้ผู้ใช้งานสามารถเข้าถึงข้อมูลนั้นได้ผ่านอินเทอร์เน็ต

โดยการสร้างฐานข้อมูล เราใช้ SQLite ในการทำ เนื่องจากฐานข้อมูลนี้อยู่ในรูปแบบตารางและ ใช้ Java Script ในการพัฒนาเว็บไซต์

3.1 Code ในการนำขึ้นเว็บไซต์

3.1.1 server.js

ส่วนแรกคือ server.js ที่เป็นการนำตารางขึ้นบนเว็บไซต์ ส่วนของการสร้างและแสดงตาราง ตัวอย่าง

```
const express = require('express');
const bodyParser = require('body-parser');
const sqlite3 = require('sqlite3').verbose();
const path = require('path');

const app = express();
const port = 3000;

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname)));
```



```

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

const db = new sqlite3.Database('./ฐานข้อมูลบริษัทน้ำเปล่า.db');

let query = `CREATE TABLE IF NOT EXISTS Product (
  Product_Id INTEGER PRIMARY KEY AUTOINCREMENT,
  Product_name TEXT,
  Product_Price NUMERIC,
  Stock NUMERIC,
  Product_cost NUMERIC
)`;

db.run(query);

app.get('/product', (req, res) => {

  let query = 'SELECT * FROM Product';

  db.all(query, (err, rows) => {
    if (err) {
      let msg = {error: err.message};
      status(500).json(msg);
    }
    res.json(rows);
  });
});

```

อีกส่วนคือ การหาข้อมูลภายในตาราง ตัวอย่าง

```

app.get('/search-product', (req, res) => {
  const conditions = [];
  const params = [];
  if (req.query.Product_Id) {
    conditions.push('Product_Id = ?');
    params.push(req.query.Product_Id);
  }
}

```

```

    }
    if (req.query.Product_name) {
        conditions.push('Product_name LIKE ?');
        params.push(`%${req.query.Product_name}%`);
    }
    if (req.query.Product_Price) {
        conditions.push('Product_Price = ?');
        params.push(req.query.Product_Price);
    }
    if (req.query.Stock) {
        conditions.push('Stock = ?');
        params.push(req.query.Stock);
    }
    if (req.query.Product_cost) {
        conditions.push('Product_cost = ?');
        params.push(req.query.Product_cost);
    }
    let query = 'SELECT * FROM Product';
    if (conditions.length > 0) {
        query += ' WHERE ' + conditions.join(' AND ');
    }
    db.all(query, params, (err, rows) => {
        if (err) {
            return res.status(500).json({ error: err.message });
        }
        res.json(rows);
    });
});
});

```

3.1.2 HTML

ส่วนที่สองคือ HTML หรือส่วนที่แสดงบนเว็บไซต์ ตัวอย่าง

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>บริษัทน้ำเปล่า</title>
<link href="style1.css" rel="stylesheet" type="text/css">
</head>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Water Company Database</title>
  <link rel="stylesheet" href="style1.css">
</head>
<body>
  <div class="container">
    
    <h1>ฐานข้อมูลบริษัทน้ำเปล่า</h1>
    <nav>
      <a href="home.html">Home</a>
      <a href="about.html">About</a>
      <a href="product.html">Product</a>
      <a href="order.html">Order</a>
      <a href="payment.html">Payment</a>
      <a href="contact.html">Contact</a>
    </nav>
    <div class="content">
      <p>ยินดีต้อนรับสู่ฐานข้อมูลของบริษัทน้ำเปล่า</p>
      <p>เลือกหมวดหมู่เพื่อดูข้อมูล</p>
    </div>
  </div>
</body>
<body>
  <h1>Product</h1>
  <form id="productForm">
    <input type="number" id="Product_Id" placeholder="Product_Id">
    <input type="text" id="Product_name" placeholder="Product_name">
    <input type="number" id="Product_Price" placeholder="Product_Price">
    <input type="number" id="Stock" placeholder="Stock">
    <input type="number" id="Product_cost" placeholder="Product_cost">
    <button type="submit">Find Product</button>
  </form>

```

```

    </form>
<table id="productList">
    <tr>
        <th>Product_Id</th>
        <th>Product_name</th>
        <th>Product_Price</th>
        <th>Stock</th>
        <th>Product_cost</th>
    </tr>
</table>
<br>
<h1>Orders</h1>
    <form id="ordersForm">
        <input type="number" id="Order_Id" placeholder="Order_Id">
        <input type="number" id="Order_Customer_Id" placeholder="Customer_Id">
        <input type="number" id="Order_Payment_Id" placeholder="Payment_Id">
        <input type="text" id="Order_Date" placeholder="Order_Date">
        <input type="text" id="Order_Status" placeholder="Order_Status">
        <button type="submit">Find Order</button>
    </form>
<table id="ordersList">
    <tr>
        <th>Order_Id</th>
        <th>Customer_Id</th>
        <th>Payment_Id</th>
        <th>Order_Date</th>
        <th>Order_Deadline</th>
        <th>Order_Month</th>
        <th>Order_Year</th>
        <th>Product_Id</th>
        <th>Order_Count</th>
        <th>Order_Price</th>
        <th>Order_Status</th>
    </tr>
</table>

```

ส่วนที่แสดงบนหน้าเว็บไซต์นี้ นอกจากส่วนของฐานข้อมูลแล้วยังมีส่วนอื่นๆ เช่น

หน้าหลัก, เกี่ยวกับ หรือ ช่องทางการติดต่อ เช่น

```
//Home หรือ หน้าหลัก
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>บริษัทน้ำเปล่า</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div class="container">
    
    <h1>WATER WATER</h1>
    <nav>
      <a href="about.html">About</a>
      <a href="product.html">Product</a>
      <a href="payment.html">Payment</a>
      <a href="order.html">Order</a>
      <a href="contact.html">Contact</a>
      <a href="table.html">Database</a>
    </nav>
    <div class="content">
      <h2>ยินดีต้อนรับ</h2>
      <p>เลือกหมวดหมู่จากเมนูเพื่อดูข้อมูลเพิ่มเติม</p>
      <h2>สินค้าแนะนำ</h2>
      <div class="recommended-products">
        <div class="product">
          
          <h3>น้ำดื่ม 500 มิลลิลิตร</h3>
          <p>ราคา: 10 บาท</p>
          <p>น้ำดื่มคุณภาพสูงในขวด PET ขนาดพกพา</p>
        </div>
        <div class="product">
```

```

        
        <h3>น้ำดื่ม 1 ลิตร</h3>
        <p>ราคา: 15 บาท</p>
        <p>น้ำดื่มบริสุทธิ์เหมาะสำหรับการบริโภคในบ้านหรือ
องค์กร</p>
    </div>
    <div class="product">
        
        <h3>น้ำดื่ม 1.5 ลิตร</h3>
        <p>ราคา: 20 บาท</p>
        <p>น้ำดื่มขนาดใหญ่เหมาะสำหรับการเดินทางหรือกิจกรรมกลางแจ้ง</p>
    </div>
</div>
<div class="info-box">
    <h2>พันธกิจของเรา</h2>
    <p>เรามุ่งมั่นที่จะเป็นผู้นำด้านผลิตภัณฑ์น้ำดื่มที่ใส่ใจสิ่งแวดล้อม ด้วยการใช้องค์กรที่สามารถรีไซเคิลได้ รวมถึงสนับสนุนการอนุรักษ์น้ำและทรัพยากรธรรมชาติ</p>
    <h2>สินค้าของเรา</h2>
    <p>บริษัทเรามีสินค้าน้ำดื่มหลายรูปแบบ เช่น น้ำดื่มขวด PET ขนาด 500 มิลลิลิตร, 1 ลิตร, และ 1.5 ลิตร เพื่อตอบสนองความต้องการของลูกค้าในทุกโอกาส ทั้งเพื่อการบริโภคส่วนตัวและการใช้งานในองค์กร</p>
    <h2>มาตรฐานและคุณภาพ</h2>
    <p>เราใช้เทคโนโลยีการกรองและฆ่าเชื้อที่ทันสมัย เพื่อให้มั่นใจได้ว่าผลิตภัณฑ์น้ำดื่มของเรามีคุณภาพที่ปลอดภัยต่อสุขภาพ ทุกขวดของน้ำดื่มผ่านการตรวจสอบโดยผู้เชี่ยวชาญก่อนถึงมือผู้บริโภค</p>
    <h2>การดูแลลูกค้า</h2>
    <p>เรามีศูนย์บริการลูกค้าที่พร้อมให้บริการทุกวัน ตั้งแต่เวลา 8.00 น. ถึง 17.00 น. โดยทีมงานของเราพร้อมให้คำแนะนำเกี่ยวกับผลิตภัณฑ์ และตอบคำถามที่เกี่ยวข้องกับการสั่งซื้อและการชำระเงิน</p>

```

```

        <h2>ร่วมเป็นส่วนหนึ่งในการอนุรักษ์สิ่งแวดล้อม</h2>
        <p>เราสนับสนุนให้ลูกค้าร่วมกันลดการใช้ขวดพลาสติกแบบ
ใช้ครั้งเดียว และหันมาใช้บรรจุภัณฑ์ที่สามารถรีไซเคิลหรือใช้ซ้ำได้ เพื่อช่วยลด
ปริมาณขยะพลาสติกในธรรมชาติ</p>
    </div>
</div>
</div>
</body>
</html>
//About หรือ เกี่ยวข้องกับเรา
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>เกี่ยวกับเรา - บริษัทน้ำเปล่า</title>
    <link href="style.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        
        <nav>
            <a href="home.html">Home</a>
            <a href="product.html">Product</a>
            <a href="order.html">Order</a>
            <a href="payment.html">Payment</a>
            <a href="contact.html">Contact</a>
            <a href="table.html">Database</a>
        </nav>
        <div class="info-box">
            <h2>เกี่ยวกับเรา</h2>
            <p>บริษัทน้ำเปล่ามีความมุ่งมั่นในการส่งมอบน้ำดื่มบริสุทธิ์ที่ดีที่สุด
ให้กับลูกค้า ด้วยเทคโนโลยีที่ทันสมัยและกระบวนการผลิตที่ได้มาตรฐาน</p>
            <h2>ประวัติของบริษัท</h2>
            <p>ก่อตั้งขึ้นในปี 2567 บริษัทน้ำเปล่ามุ่งมั่นที่จะผลิตน้ำดื่มคุณภาพ
สูง เพื่อสนับสนุนความต้องการของลูกค้าในทุกกลุ่มเป้าหมาย ด้วยการใช้วัตถุดิบที่

```

```

    ดีที่สุดและการผลิตที่ปลอดภัย</p>
        <h2>วิสัยทัศน์</h2>
        <p>เรามีวิสัยทัศน์ในการเป็นผู้นำด้านการผลิตน้ำดื่มที่มีคุณภาพ
    สูงและใส่ใจต่อสิ่งแวดล้อม โดยเรามีการลงทุนในเทคโนโลยีที่เป็นมิตรต่อสิ่ง
    แวดล้อม และใช้บรรจุภัณฑ์ที่สามารถรีไซเคิลได้</p>
        <h2>ทีมงานของเรา</h2>
        <p>เรามีทีมงานที่มีความเชี่ยวชาญและประสบการณ์ในการผลิต
    น้ำดื่มอย่างมืออาชีพ ที่พร้อมให้บริการและแนะนำผลิตภัณฑ์น้ำดื่มที่เหมาะสมกับ
    ความต้องการของลูกค้า</p>
        
    </div>
</div>
</body>
</html>
//Contact หรือ ช่องทางการติดต่อ
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ติดต่อเรา - บริษัทน้ำเปล่า</title>
    <link href="style.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        
        <nav>
            <a href="home.html">Home</a>
            <a href="about.html">About</a>
            <a href="product.html">Product</a>
            <a href="order.html">Order</a>
            <a href="payment.html">Payment</a>
            <a href="table.html">Database</a>
        </nav>
        <div class="content">
            <h1>ติดต่อเรา</h1>

```



```

    <p>ที่อยู่: 123 ตำบลคลองหก อำเภอคลองหลวง จังหวัด
ปทุมธานี</p>
    <p>เบอร์ติดต่อ: 02-123-4567</p>
    <p>อีเมล: waterwater@watercompany.com</p>
    
    <p>Facebook: WATER WATER</p>
    
    <p>Line: @WATER WATER</p>
    
    <p>Instagram: WATER_WATER</p>
  </div>
</div>
</body>
</html>

```

3.1.3 script.js

ส่วนที่สามคือ script.js ตัวอย่าง

```

async function fetchProducts() {
  const response = await fetch('/product');
  const products = await response.json();
  displayProducts(products);
}

function displayProducts(products) {
  const productList = document.getElementById('productList');
  productList.innerHTML = `
<tr>
<th>Product_Id</th>
<th>Product_name</th>
<th>Product_Price</th>
<th>Stock</th>
<th>Product_cost</th>
</tr>`;

  products.forEach(product => {

```

```

        productList.innerHTML += `
<tr>
<td>${product.Product_Id}</td>
<td>${product.Product_name}</td>
<td>${product.Product_Price}</td>
<td>${product.Stock}</td>
<td>${product.Product_cost}</td>
</tr>`;
    });
}
// Search event listeners
document.getElementById('productForm').addEventListener('submit', async (e) => {
    e.preventDefault();
    const searchParams = {
        Product_Id: document.getElementById('Product_Id').value,
        Product_name: document.getElementById('Product_name').value,
        Product_Price: document.getElementById('Product_Price').value,
        Stock: document.getElementById('Stock').value,
        Product_cost: document.getElementById('Product_cost').value
    };
    const response = await fetch('/search-product?' + new URLSearchParams(
        Object.entries(searchParams).filter(([_, value]) => value !
== ''))
    ));
    const products = await response.json();
    displayProducts(products);
});
fetchProducts();

```

บทที่ 4

Data On GIT