

รายงานประกอบรายวิชา
เทคโนโลยีการเขียนโปรแกรมเว็บไซต์
Web Programming Technology
รหัสวิชา 226297

เสนอ เว็บแอปพลิเคชัน
ระบบจัดการสต็อกสินค้าผ่านเว็บไซต์

ผู้จัดทำ

65024658 นายพชร พรหมพฤษ์จรัส

65024726 นายสรารุท ค่ำสด

65021981 นายภควัฒน์ ใจลังกา

65021363 นายกิตติพงษ์ ปรีดาภิรัตน์

ผู้สอน

อาจารย์ วัฒนพงศ์ สุทธภักดิ์

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศ
และการสื่อสาร มหาวิทยาลัยพะเยา
ภาคการศึกษา ปลาย ประจำปีการศึกษา 2566

บทที่ 1

ที่มาของเว็บไซต์

1. ที่มาและความสำคัญ

เนื่องจากระบบจัดการฐานข้อมูลมีหลากหลายและอาจจะมีผู้ที่ผลิตขึ้นมานั้นมากมาย แต่ก็มีโอกาสที่จะมีรูปแบบซ้ำกันได้ มีความซับซ้อน มีฟังก์ชันที่ไม่จำเป็นต้องใช้งานและไม่ตรงกับความต้องการของผู้ใช้งาน หรือไม่มีความเป็นเอกลักษณ์ ทั้งนี้ทางกลุ่มเราต้องการความเป็นเอกลักษณ์ในระบบการจัดการฐานข้อมูล จึงมีการเขียนโปรแกรมนี้ขึ้นมาเพื่อให้ตรงกับความต้องการของผู้ใช้งานมากขึ้นและมีความเป็นเอกลักษณ์เฉพาะ และมีความง่ายต่อการใช้งาน แม้ผู้ที่ไม่ถนัดด้านคอมพิวเตอร์ก็สามารถใช้งานได้อย่างง่ายดาย

2. วัตถุประสงค์

- เพิ่มประสิทธิภาพในการจัดการข้อมูล
เก็บข้อมูลรถยนต์ รายละเอียด สภาพ ราคา รูปภาพ ข้อมูลลูกค้า การติดต่อ นัดหมาย ช่อม บำรุง อะไหล่ ได้อย่างเป็นระบบ วิเคราะห์ข้อมูล สถิติ ยอดขาย รุ่นรถ ยี่ห้อ ความนิยม ช่วย ให้การทำงานสะดวก ประหยัดเวลา
- ขยายช่องทางการขายและเข้าถึงลูกค้า
โปรโมทรถมือสองบนเว็บไซต์ เพิ่มช่องทางการเข้าถึงลูกค้า โดยลงรายละเอียดรถ รูปภาพ ข้อมูลครบถ้วน ช่วยให้ลูกค้าตัดสินใจเลือกซื้อได้ง่าย
- ลดต้นทุนและเพิ่มผลกำไร
ประหยัดค่าใช้จ่ายในการจัดการข้อมูล พิมพ์เอกสาร จ้างพนักงาน เพิ่มประสิทธิภาพ การทำงาน ประหยัดเวลา ค่าใช้จ่าย

3. ประโยชน์ที่ได้รับ

- ควบคุมสินค้าคงคลังได้อย่างมีประสิทธิภาพ
ป้องกันปัญหาสินค้าขาดสต็อก หรือสินค้าล้นเกิน
- บริหารจัดการสินค้าได้อย่างมีประสิทธิภาพ
จัดกลุ่มสินค้าตามแบรนด์ หรือรุ่น
ทำให้ค้นหาสินค้าได้อย่างรวดเร็ว
ตรวจสอบประวัติการขายสินค้า
วิเคราะห์สินค้าที่ขายดี สินค้าที่ขายได้ช้า

- ประหยัดเวลาและต้นทุน
 - ลดการใช้แรงงานคน
 - ลดความผิดพลาดจากการจัดการสินค้า
 - เพิ่มประสิทธิภาพในการทำงาน
 - ประหยัดค่าใช้จ่าย
- เพิ่มขีดความสามารถในการวิเคราะห์ข้อมูล
 - วิเคราะห์ข้อมูลการขายสินค้า
 - ตัดสินใจทางธุรกิจได้อย่างมีประสิทธิภาพ
- เพิ่มประสิทธิภาพในการทำงาน
- รองรับการเติบโตของธุรกิจ
 - รองรับการใช้งานบนอุปกรณ์ต่างๆ

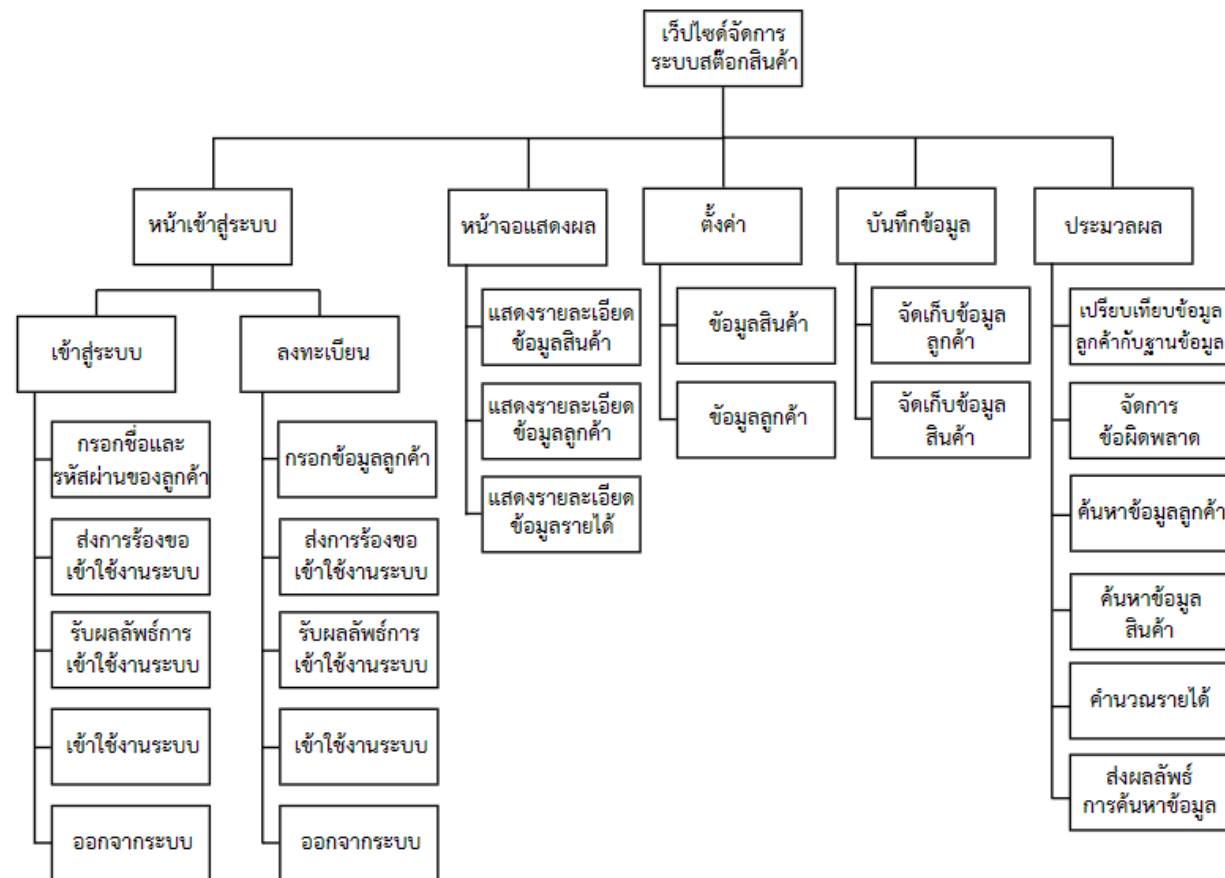
4. เทคโนโลยีที่ใช้

- Visual Studio Code
- Chat GPT
- Google Bard Ai
- Black Box

บทที่ 2

โครงสร้างเว็บไซต์

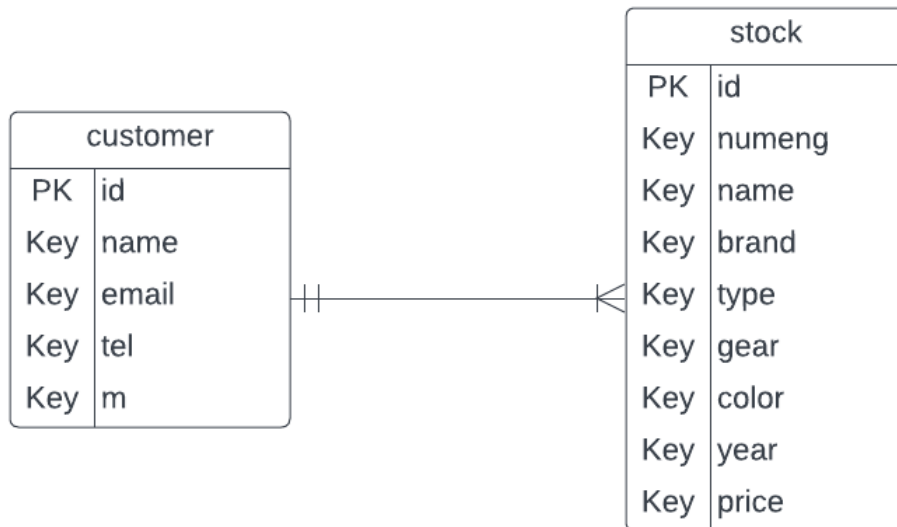
แสดงข้อมูลโครงสร้างเว็บไซต์ ด้วย Tree Diagram



บทที่ 3

ข้อมูลฐานข้อมูล

แสดงข้อมูล Database Schema หรือ ER Diagram



บทที่ 4

การติดตั้งและใช้งานเว็บไซต์

1. การติดตั้ง

อธิบายลำดับการติดตั้งเว็บไซต์ ตั้งแต่การ clone git repository ที่เก็บเว็บไซต์ ไปจนถึงการ build-run website

- นำไฟล์ test.sql ลง database

- เรียกใช้งานคำสั่งผ่าน terminal ของ vscode

```
npm i express mysql2 bcryptjs body-parser
```

```
node index.js
```

- เข้าลิ้งตามต้งไปนี้เพื่อใช้งาน

http://localhost:3308/login_admin

2. การเรียกใช้งาน

อธิบายการเปิดเว็บไซต์เรียกดูเว็บโปรเจค พร้อม capture รูปภาพหน้าจอเว็บไซต์ พร้อมทั้งกำกับฟังก์ชันผู้รับผิดชอบ

2.1 หลังบ้าน

ดึงข้อมูลสินค้าจาก database ตำแหน่ง stock มาแสดงที่หน้า web

```
app.get('/data', (req, res) => {
  connection.query('SELECT * FROM stock', (error, results, fields) => {
    if (error) throw error;
    res.json(results);
  });
});
```

```
<script>
  fetch('/data')
  .then(response => response.json())
  .then(data => {
    // เรียงข้อมูลตามลำดับ id (จากน้อยไปมาก)
    data.sort((a, b) => a.id - b.id);

    const carTableBody = document.getElementById('car-table-body');
    data.forEach(car => {
      const row = document.createElement('tr');
      row.innerHTML = `
        <td>${car.id}</td>
        <td>${car.numeng}</td>
        <td>${car.name}</td>
        <td>${car.brand}</td>
        <td>${car.type}</td>
        <td>${car.gear}</td>
        <td>${car.color}</td>
        <td>${car.year}</td>
        <td>${car.price}</td>
      `;
      carTableBody.appendChild(row);
    });
  });
};
```

กำหนด username และ password ในการเข้าถึงการใช้งาน

```
////////////////////////////////////
const users = [
  { username: 'admin', password: '123' },
  { username: 'owner', password: '123' },
  { username: 'son', password: '123' },
];

app.post('/login', (req, res) => {
  const username = req.body.username;
  const password = req.body.password;

  const user = users.find(
    (user) => user.username === username && user.password === password
  );

  if (user) {
    res.redirect('/stock');
  } else {
    res.send('Invalid username or password');
  }
});

<script>
const loginForm = document.getElementById('login-form');
const users = [
  { username: 'admin', password: '123' },
  { username: 'owner', password: '123' },
  { username: 'son', password: '123' },
];

loginForm.addEventListener('submit', (event) => {
  event.preventDefault();
  const username = document.getElementById('username').value;
  const password = document.getElementById('password').value;

  const user = users.find(
    (user) => user.username === username && user.password === password
  );

  if (user) {
    alert('Login successful');
    window.location.href = 'index2.html';
  } else {
    alert('Invalid username or password');
  }
});
</script>
```

เลือกใช้งาน id ของhtml ที่จะดึงข้อมูลมาบันทึกและกำหนดตำแหน่งที่จะบันทึก ส่วน stock

```
app.post('/insert-stock', (req, res) => {
  const numeng = req.body.numeng;
  const name = req.body.name;
  const year = req.body.year;
  const type = req.body.type;
  const color = req.body.color;
  const brand = req.body.brand;
  const seat = req.body.seat;
  const gear = req.body.gear;
  const price = req.body.price;

  // create an array of values to insert into the database
  const values = [numeng, name, year, type, color, brand, seat, gear, price];

  // build the SQL query using placeholders for the values
  const query = 'INSERT INTO stock (numeng, name, year, type, color, brand, seat, gear, price) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)';

  // execute the query with the values
  connection.query(query, values, (error, results, fields) => {
    if (error) throw error;
    res.redirect('/stock');
  });
});
```

รับค่าตัวเลขจาก tag html แล้วเลือกข้อมูลที่จะลบโดยอ้างอิงจากเลข id ส่วน stock

```
app.post('/delete-stock-tag', (req, res) => {
  const tagId = req.body.tagId;

  // build the SQL query to delete the row with the given tag_id
  const query = 'DELETE FROM stock WHERE id = ?';

  // execute the query with the tag_id value
  connection.query(query, [tagId], (error, results, fields) => {
    if (error) throw error;

    // check if any rows were affected
    if (results.affectedRows > 0) {
      // if the delete was successful, redirect to the stock page
      res.redirect('/stock');
    } else {
      // if no rows were affected, display an error message
      res.status(404).send('Tag not found');
    }
  });
});
```


ดึงข้อมูลจาก database รายการสินค้าที่ขายไปแล้ว โดยอ้างอิงจาก tag Id

```
app.post('/save-data', (req, res) => {
  const id = req.body.id;

  // ดึง SQL query ดึงข้อมูลของ test.stock โดยใช้ ID
  const selectQuery = 'SELECT * FROM test.stock WHERE id = ?';

  // Execute the query with the ID value
  connection.query(selectQuery, [id], (selectError, selectResults) => {
    if (selectError) throw selectError;

    // Check if any rows were found
    if (selectResults.length > 0) {
      // ดึง SQL query สำหรับ insert ข้อมูล test.sell
      const insertQuery = 'INSERT INTO test.sell (puugggg, name, brand, seat, gear, color, year, price, type) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)';

      // Adjust the values accordingly based on the columns in test.sell
      const valuesToInsert = [
        selectResults[0].numerg,
        selectResults[0].name,
        selectResults[0].brand,
        selectResults[0].seat,
        selectResults[0].gear,
        selectResults[0].color,
        selectResults[0].year,
        selectResults[0].price,
        selectResults[0].type
      ];

      // Insert the data into the test.sell table
      connection.query(insertQuery, valuesToInsert, (insertError, insertResults) => {
        if (insertError) throw insertError;
        res.status(200).send('Data saved successfully');
      });
    } else {
      res.status(404).send('Stock not found');
    }
  });
});
```

ลบข้อมูล database ที่ตำแหน่ง stock โดยอ้างอิงผ่าน id

```
app.post('/delete-stock-tag', (req, res) => {
  const tagId = req.body.tagId;

  // build the SQL query to delete the row with the given tag_id
  const query = 'DELETE FROM stock WHERE id = ?';

  // execute the query with the tag_id value
  connection.query(query, [tagId], (error, results, fields) => {
    if (error) throw error;

    // check if any rows were affected
    if (results.affectedRows > 0) {
      // if the delete was successful, redirect to the stock page
      res.redirect('/stock');
    } else {
      // if no rows were affected, display an error message
      res.status(404).send('Tag not found');
    }
  });
});
```

เลือกใช้งาน id ของhtml ที่จะดึงข้อมูลมาบันทึกและกำหนดตำแหน่งที่จะบันทึก ส่วนของ customer

```
app.post('/insert-customer', (req, res) => {
  const name = req.body.name;
  const email = req.body.email;
  const tel = req.body.tel;
  const money = req.body.money;

  // Create an SQL query to insert the customer data into the database
  const query = 'INSERT INTO customer (name, email, tel, money) VALUES (?, ?, ?, ?)';
  connection.query(query, [name, email, tel, money], (error, results, fields) => {
    if (error) throw error;

    // Redirect the user to the customer list page
    res.redirect('/customer-list');
  });
});
```

ดึงข้อมูลสินค้าจาก database ตำแหน่ง customer มาแสดงที่หน้า web

```
app.get('/customer-data', (req, res) => {
  // Query the database for customer data
  const query = 'SELECT * FROM customer';
  connection.query(query, (error, results, fields) => {
    if (error) throw error;

    // Return the customer data as a JSON response
    res.json(results);
  });
});
```

```
</script>
<script>
  fetch('/customer-data')
    .then(response => response.json())
    .then(data => {
      // Sort data by ID
      data.sort((a, b) => a.id - b.id);

      const customerTableBody = document.getElementById('customer');
      data.forEach(customer => {
        const row = document.createElement('tr');
        row.innerHTML = `
          <td>${customer.id}</td>
          <td>${customer.name}</td>
          <td>${customer.email}</td>
          <td>${customer.tel}</td>
          <td>${customer.money}</td>
        `;
        customerTableBody.appendChild(row);
      });
    });
</script>
```

ดึงข้อมูลสินค้าจาก database ตำแหน่ง sell มาแสดงที่หน้า web

```
app.get('/data-sell', (req, res) => {
  connection.query('SELECT * FROM test.sell', (error, results, fields) => {
    if (error) throw error;
    res.json(results);
  });
});
```

```
<script>
  fetch('/data-sell')
    .then(response => response.json())
    .then(data => {
      // ดึงข้อมูลจาก id (300 ไม่อยู่ 1000)
      data.sort((a, b) => a.id - b.id);

      const carTableBody = document.getElementById('sell-ta');
      data.forEach(car => {
        const row = document.createElement('tr');
        row.innerHTML = `
          <td>${car.id}</td>
          <td>${car.guugoo}</td>
          <td>${car.name}</td>
          <td>${car.brand}</td>
          <td>${car.type}</td>
          <td>${car.gear}</td>
          <td>${car.color}</td>
          <td>${car.year}</td>
          <td>${car.price}</td>
        `;
        carTableBody.appendChild(row);
      });
    });
</script>
```

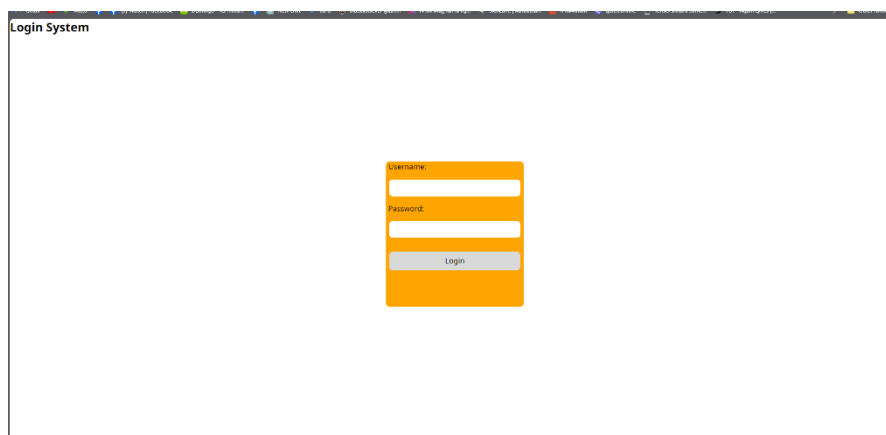
2.2 หน้าบ้าน

- หน้า login

ผู้ใช้อกรอก username และ password ตามที่กำหนดไว้ เนื่องจากเป็นระบบส่วนผู้ดูแลระบบซึ่งได้มีการกำหนด username และ password ไว้แล้วเพื่อความปลอดภัยที่บุคคลภายนอกจะไม่สามารถเข้ามาใช้งานได้ โดยให้ใช้ username และ password ดังต่อไปนี้เพื่อเข้าสู่ระบบ

Username : admin

Password : 123

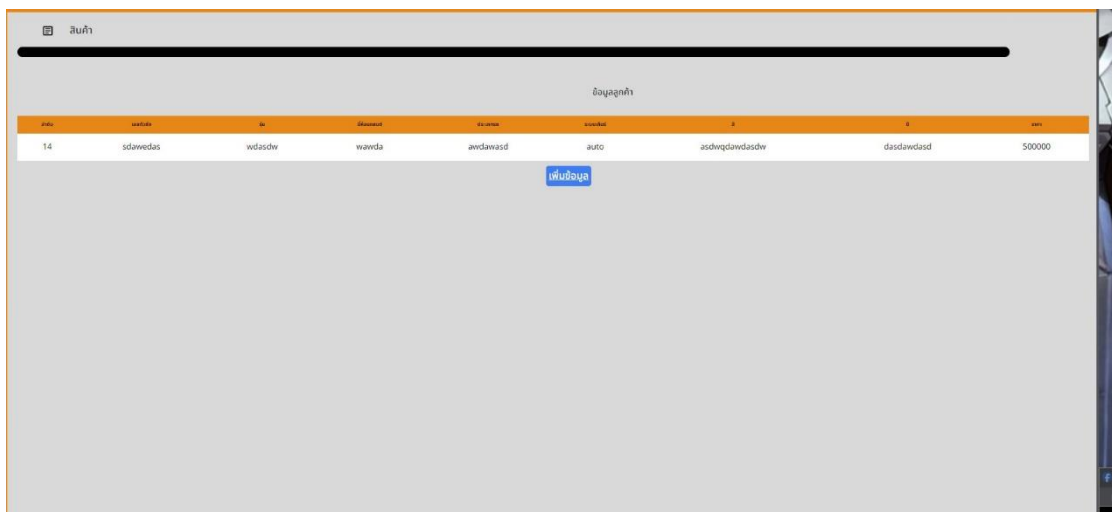


เมื่อทำการ login เข้าสู่ระบบแล้ว จะเข้าถึงระบบการใช้งานหลัก

- หน้าสินค้า

ในหน้าสินค้าจะแสดงข้อมูลของลูกค้า โดยมีฟังก์ชันอยู่ซ้ายเป็นรูปแบบปุ่ม ซึ่งมีข้อมูลที่แสดงมีดังต่อไปนี้ :

- ลำดับ
- เลขตัวถัง
- รุ่น
- ยี่ห้อรถยนต์
- ประเภท
- ระบบเกียร์
- สี
- ปี
- ราคา



index	name	name	name	name	name	name	name	price
14	sdawedas	wdssdw	wawda	awdanasd	auto	asdwqdasdsw	dardawasd	500000

เพิ่มข้อมูล

- หน้าบันทึกข้อมูล

เมื่อต้องการบันทึกข้อมูลจะต้องกรอกดังต่อไปนี้

- เลขตัวถัง
- รุ่น
- ปี
- ประเภท
- สี
- ยี่ห้อรถยนต์

- จำนวนที่นี้่ง
- ระบบเกียร์
- ราคา

อีกส่วนหนึ่งคือการบันทึกข้อมูลการขายผ่าน ID และ ลบข้อมูลที่อยู่ในคลังผ่าน ID

- หน้าข้อมูลลูกค้า

หน้าข้อมูลลูกค้าจะแสดงข้อมูลลูกค้าดังต่อไปนี้

- หมายเลข
- ชื่อลูกค้า
- email
- เบอร์โทร
- จำนวนเงิน

และมีปุ่มเพิ่มข้อมูลลูกค้าเมื่อกดปุ่มจะไปหน้าบันทึกข้อมูลลูกค้า

ข้อมูลลูกค้า				
id	name	email	phone	price
0	dadad	sdad@gmail.com	asdad	2500.00
0	boss	boss123@hotmail.com	0942148915	2500.00
0	Bossnaru	Boss123@gmail.com	097184918	2500.00
0	B	Adad@hotmail.com	097184318	2500.00
0	PL	PLsaw1213@gmail.com	0975414841	5000.00

เพิ่มข้อมูล


- หน้าบันทึกข้อมูลลูกค้า

เมื่อต้องการบันทึกข้อมูลลูกค้าต้องบันทึกข้อมูลดังต่อไปนี้

- ชื่อ
- email
- เบอร์โทร
- จำนวนเงิน

*** จำเป็นต้องกรอกข้อมูลในครบถ้วนเพื่อให้ระบบสามารถบันทึกข้อมูลได้

เมื่อกดบันทึกข้อมูลจะกลับไปหน้าข้อมูลลูกค้า



The image shows a web form titled "Insert Customer". It contains four input fields with labels "Name:", "Email:", "Tel:", and "Money:" on the left. To the right of these labels are four empty text input boxes. Below the input boxes is a button labeled "Insert". The form is set against a light gray background.

ฟังก์ชันผู้รับผิดชอบ

- นายพร พรมพฤษจรัส : หน้าสินค้า
- นายสรายุทธ คำสด : หน้าบันทึกข้อมูล , หน้าบันทึกข้อมูลลูกค้า
- นายภควัฒน ใจลังกา : หลังบ้าน back-end
- นายกิตติพงษ์ ปรีดาภิรัตน์ : หน้า login