**ช่วงเช้า**

**เตรียมความพร้อม**

1. (visual studio code) editer ใช้ในการเขียนโค้ด
2. (xampp) หรือ (appserv) ใช้ในการจำลอง server

**Extensions** visual studio code

1. Auto Complete Tag
2. Auto Rename Tag
3. Format HTML in PHP
4. HTML Snippet
5. PHP Extension Pack

**พื้นฐาน PHP ตอน PHP OOP**

PHP เขียนแบบ OOP (Object-Oriented Programming) คือ การเขียนโปรแกรมในระบบเชิงวัตถุ และ แตกต่างจาก (Structure Programming) อยู่มาก Structure Programming คือกลุ่มโครงสร้าง การทำงานแบบตามลำดับ, การเลือกกระทำตามเงื่อนไข, การทำซ้ำ

**ข้อดีของการเขียน PHP OOP**

* OOP ทำได้เร็วและง่ายขึ้น
* OOP ให้โครงสร้างที่ชัดเจนสำหรับโปรแกรม
* OOP ช่วยให้โค้ด PHP แห้ง "อย่าทำซ้ำตัวเอง" และทำให้โค้ดดูแลรักษา แก้ไข และแก้ปัญหาได้ง่ายขึ้น
* OOP ทำให้สามารถสร้างแอปพลิเคชันที่นำกลับมาใช้ใหม่ได้โดยใช้โค้ดน้อยลงและใช้เวลาในการพัฒนาที่สั้นลง

**ข้อเสียของการเขียน PHP OOP**

* ซับซ้อนกว่าแบบ เชิงโครงสร้าง

**ออบเจ็กต์ (Objects) คืออะไร**

วัตถุหรืออ็อบเจ็กต์ที่มีทั้งข้อมูลและฟังก์ชันในตัวมันเอง เช่น รถคันหนึ่งในโรงรถ คือ ออบเจ็คหนึ่งๆ ซึ่งมีข้อมูลเช่น ยี่ห้อ สีรถ รุ่นที่ ขนาดหรือนำหนัก เป็นต้น จากนั้น มีการเคลื่อนของรถ การเปิดประตู นั้นคือ เมธอด (method)

องค์ประกอบของออบเจ็คมี 2 ตัว คือ ข้อมูลรายละเอียด(Object Properties) และเมธอด (Method)

class car {

    private $brand = "Honda";

    private $color = "เขียว";

    private $model = "Civic";

    public function getmyCar(){

        return "รถรุ่น {$this->brand} สี{$this->color} รุ่น {$this->model}";

    }

}

**สร้างออบเจ็กต์**

ลำดับแรกเราต้องทำการออกแบบ Template หรือที่เรียกว่าคลาส(Class)ซึ่งภายในคลาสจะประกอบไปด้วย ฟังก์ชั่นย่อยๆ และรายละเอียดของออบเจ็กต์แต่ล่ะตัวที่สามารถกำหนดได้ตามเงื่อนไขของผู้สร้าง class รูปแบบ

class car {

}

**การเรียกใช้งาน class**

//แบบที่ 1 เรียกใช้งานแบบ new คีย์เวิร์ด

$obj = new car();

echo $obj->getmyCar();

//แบบที่ 2 เรียกใช้งานแบบ static คีย์เวิร์ด

echo car2::getmyCar();

**การใช้งาน construct**

ฟังก์ชันนี้จะเรียกใช้งานอัตโนมัติเมื่อเมื่อเรียกใช้งานวัตถุจากคลาส สังเกตว่าฟังก์ชัน construct เริ่มต้นด้วยขีดล่างสองตัวติดกัน \_\_

    function \_\_construct() {

    }

**การสืบทอด class**

คลาสที่สืบทอดมาถูกกำหนดโดยใช้ extends เป็นคีย์เวิร์ด

class car extends car2{

    private $brand = "Honda";

    private $color = "เขียว";

    private $model = "Civic";

    public function getData(){

        return "{$this->getmyCar1()} <br> " . self::getmyCar2();

    }

    private function getmyCar1(){

        return "รถรุ่น {$this->brand} สี{$this->color} รุ่น {$this->model}";

    }

}

class car2 {

    static private $brand = "Suzuki";

    static private $color = "แดง";

    static private $model = "SWIFT";

    static public function getmyCar2(){

        return 'รถรุ่น '. self::$brand .' สี'. self::$color .' รุ่น '. self::$model;

    }

}

$obj = new car();

echo $obj->getData();

$obj = null;

**ช่างบ่าย**

1. สร้าง class connect database

<?php

class dbconnect{

    static private function conn(){

        $servername = "localhost";

        $username = "root";

        $password = "";

        $db = 'database name';

        $conn = new PDO("mysql:host=$servername;dbname=$db", $username, $password, array(PDO::MYSQL\_ATTR\_INIT\_COMMAND => 'SET NAMES \'UTF8\''));

        $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

        return $conn;

    }

    //รูปแบบการเรียกใช้งาน!

    //sql \* form where ?,?

    //array('data','data')

    static public function ExecuteData($sql, $data){

        $sth = self::conn()->prepare($sql);

        $sth->execute($data);

        return true;

    }

    static public function getExecute($sql, $data){

        $sth = self::conn()->prepare($sql);

        $sth->execute($data);

        return $sth->fetchAll(PDO::FETCH\_ASSOC);

    }

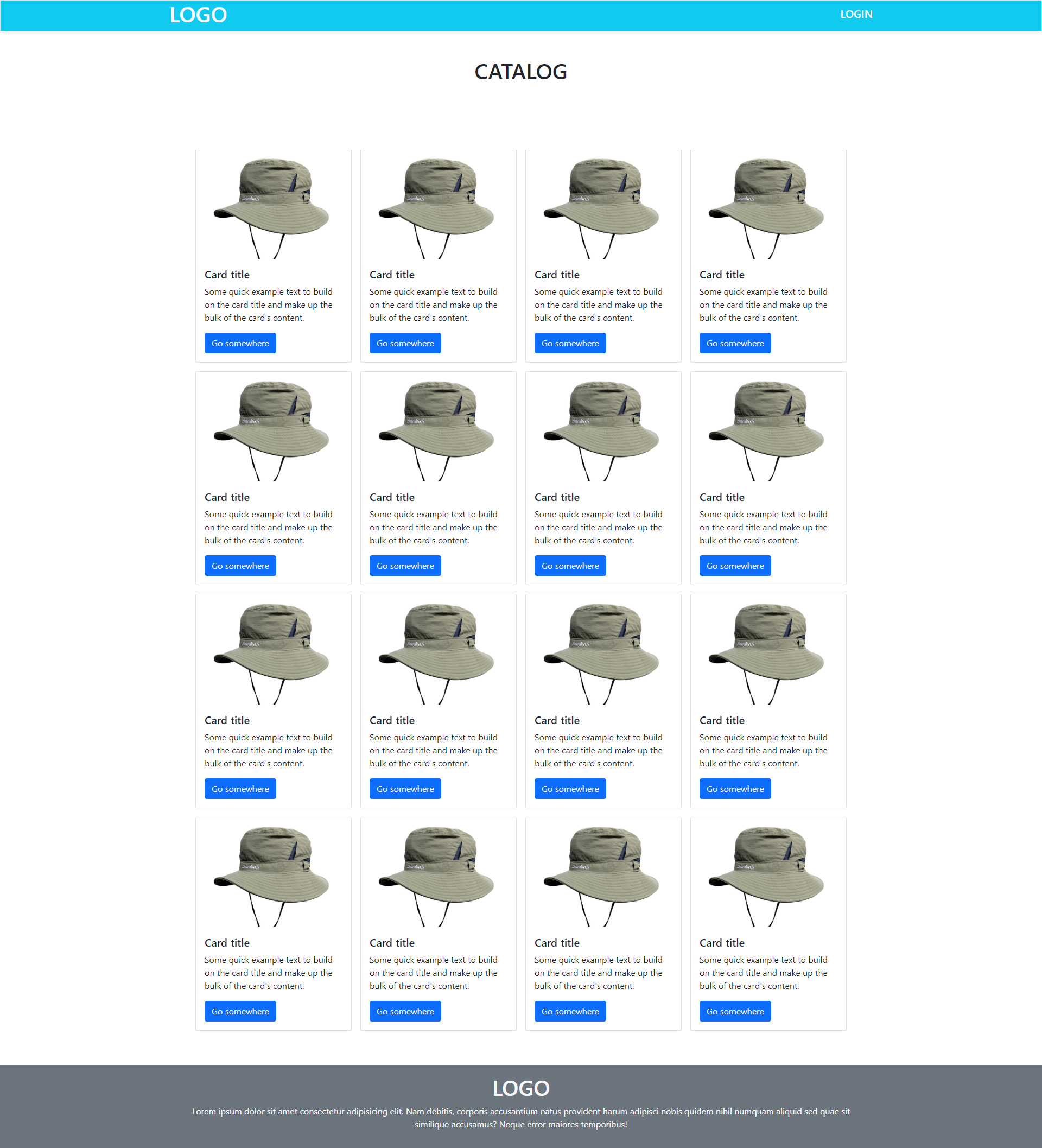
}

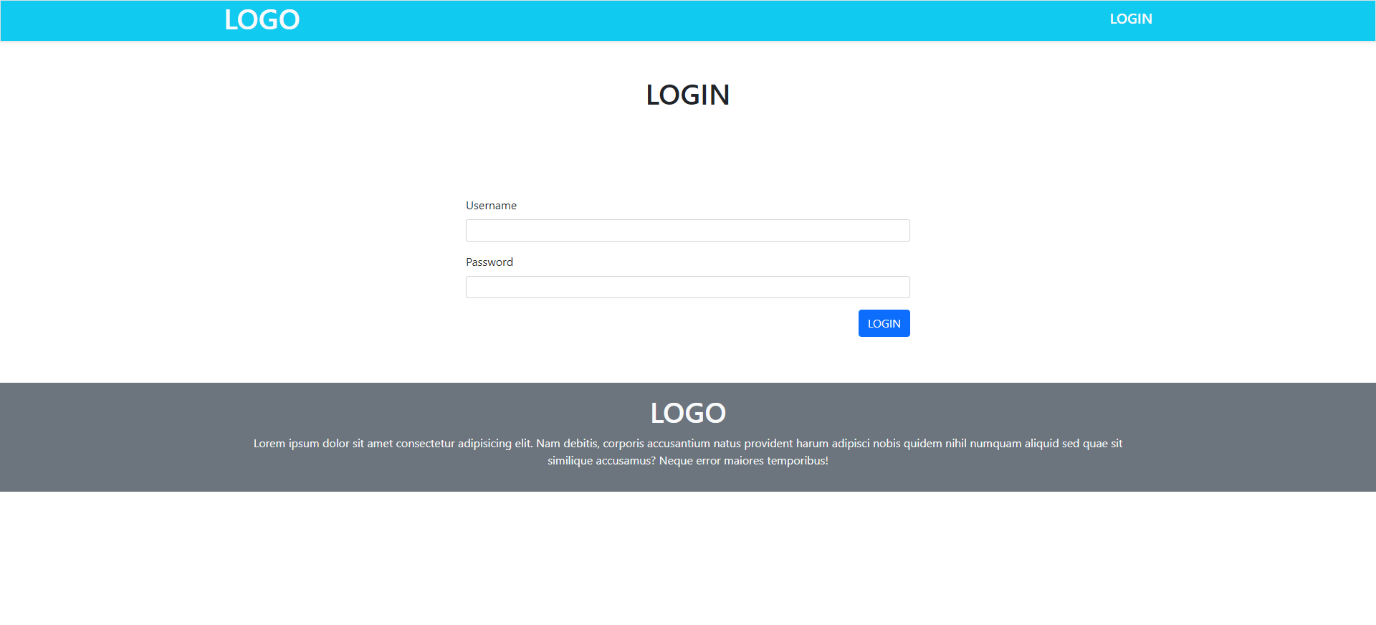
2. ลิงค์เทรมเพลตมินิโปรเจค ระบบแคตตาล็อก <https://github.com/saknarin229/_template.git>

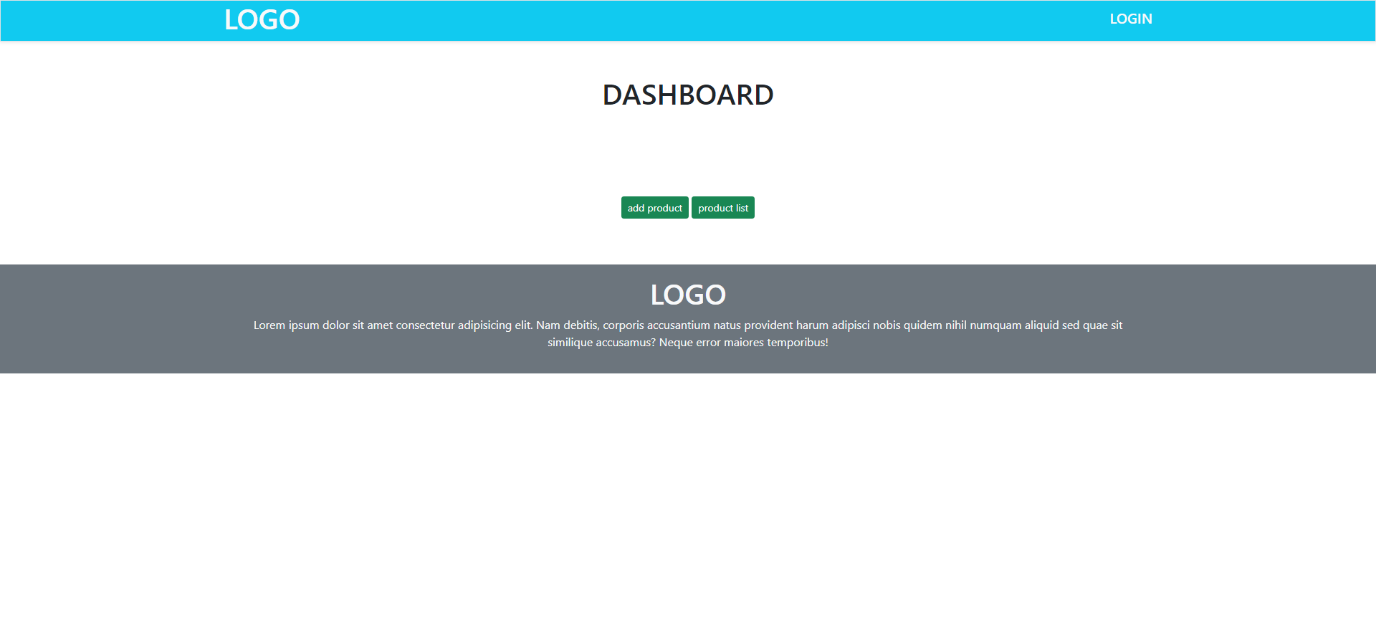
3. ออกแบบและสร้างฐานข้อมูล

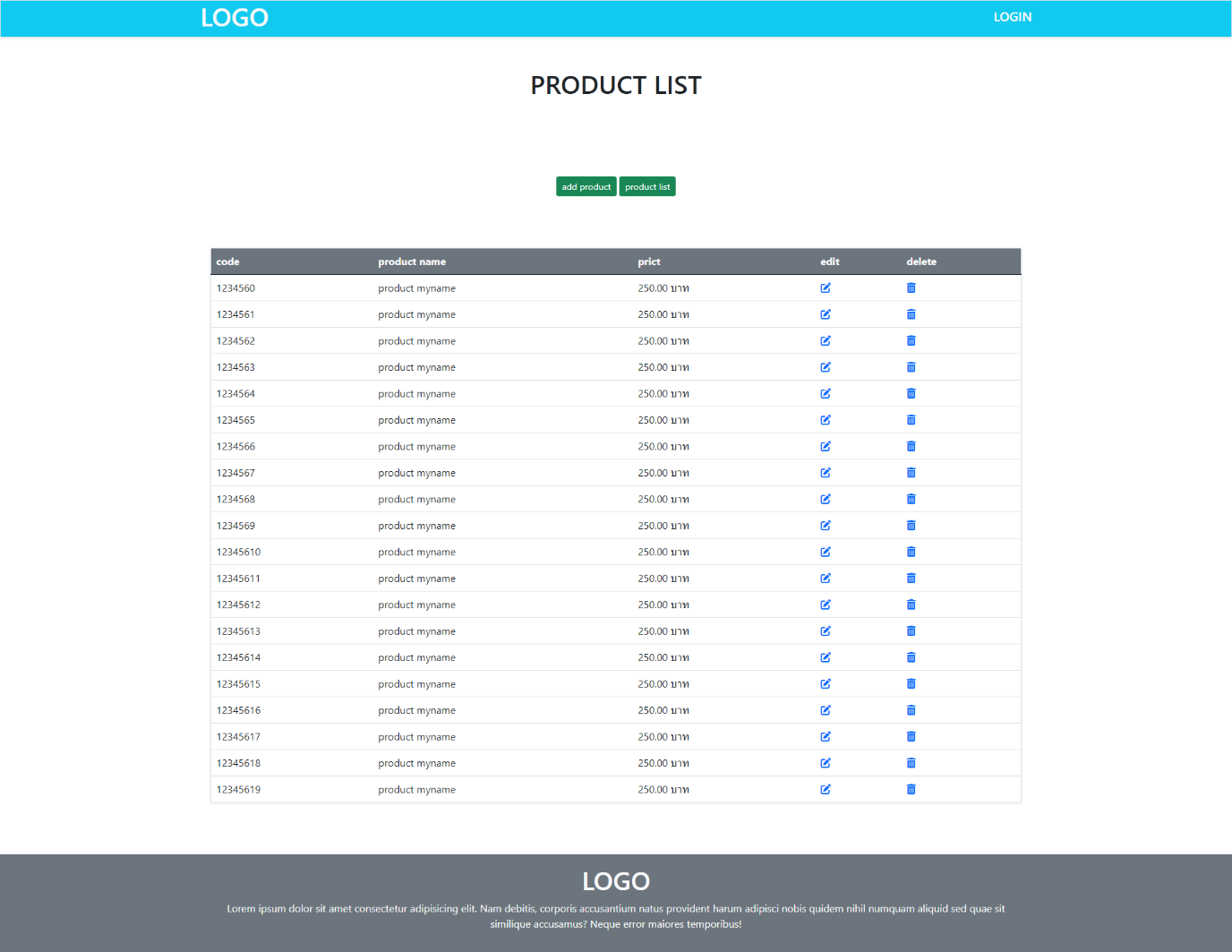
4. สร้างมินิโปรเจค ระบบแคตตาล็อก

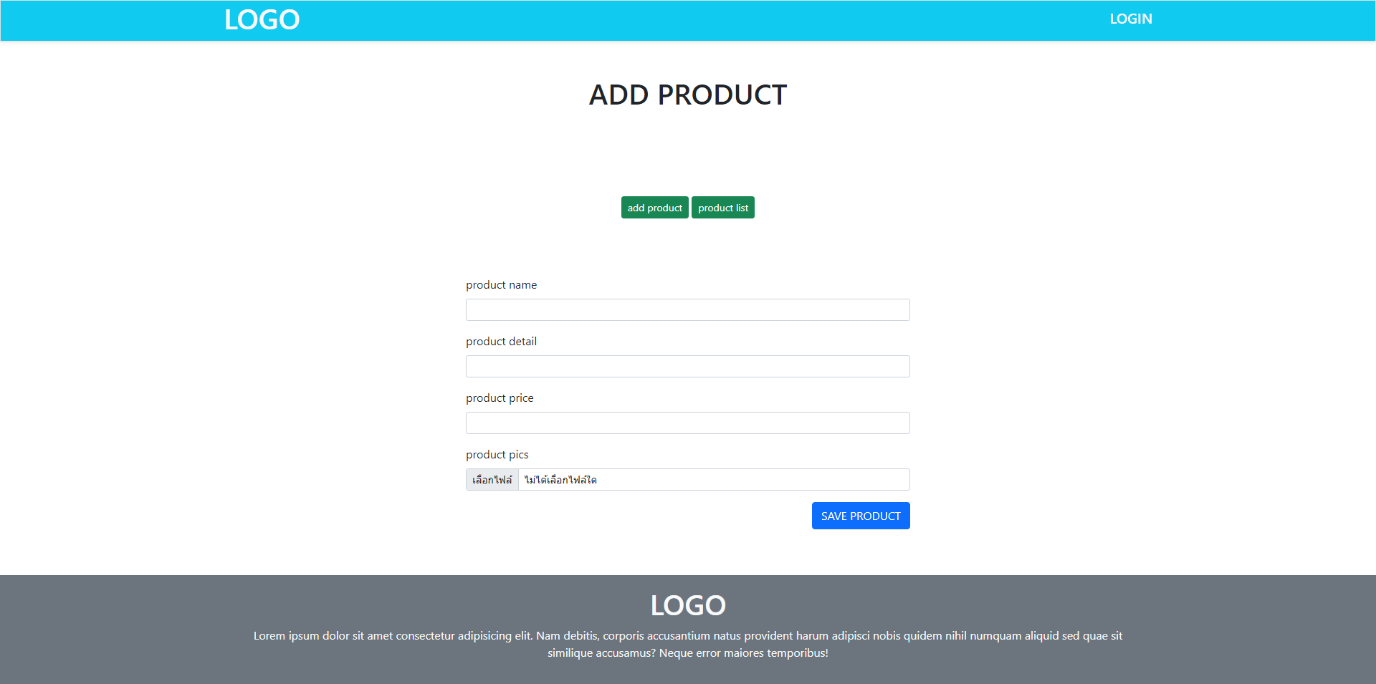
**ตัวอย่างหน้าเว็บ ระบบแคตตาล็อก**

****

****

****

****

****

**อ้างอิง https://www.w3schools.com/php/php\_oop\_inheritance.asp**