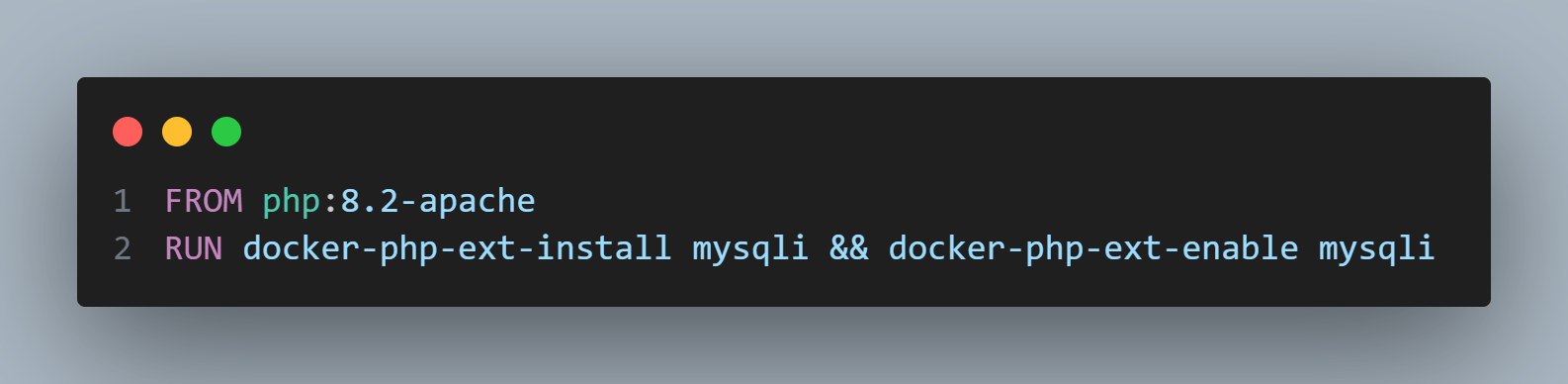
**ส่วนที่ 1: การตั้งค่าระบบจำลอง (Server Configuration)**

**1. ไฟล์ Dockerfile**

**หน้าที่หลัก:** ใช้สำหรับสร้าง Image ของ Web Server

**รายละเอียดการแก้ไข:**

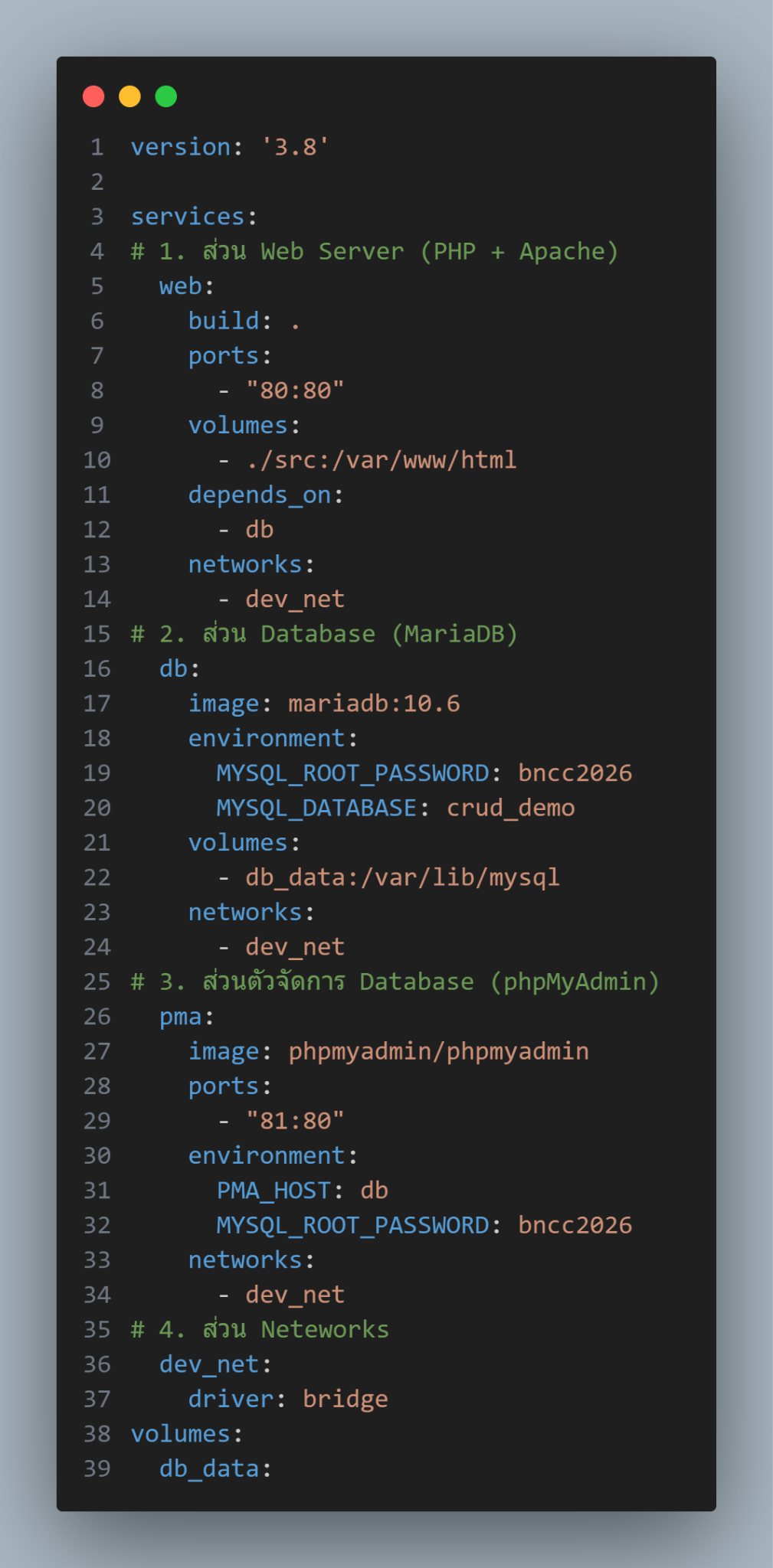
* เลือกใช้ Base Image เป็น **PHP 8.2 with Apache** เพื่อให้รองรับฟีเจอร์ใหม่ๆ ของภาษา PHP
* เพิ่มคำสั่ง **RUN docker-php-ext-install mysqli** เพื่อติดตั้งส่วนเสริม (Extension) ที่ชื่อว่า mysqli เนื่องจาก Image ปกติของ PHP จะไม่มีส่วนนี้มาให้ ทำให้ไม่สามารถเชื่อมต่อกับฐานข้อมูล MySQL หรือ MariaDB ได้
* สั่ง **docker-php-ext-enable mysqli** เพื่อเปิดใช้งานส่วนเสริมดังกล่าวให้พร้อมทำงานทันทีที่เริ่มระบบ

****

**2. ไฟล์ docker-compose.yml**

**หน้าที่:** ไฟล์ Configuration หลักสำหรับบริหารจัดการ Container และระบบเครือข่ายจำลอง

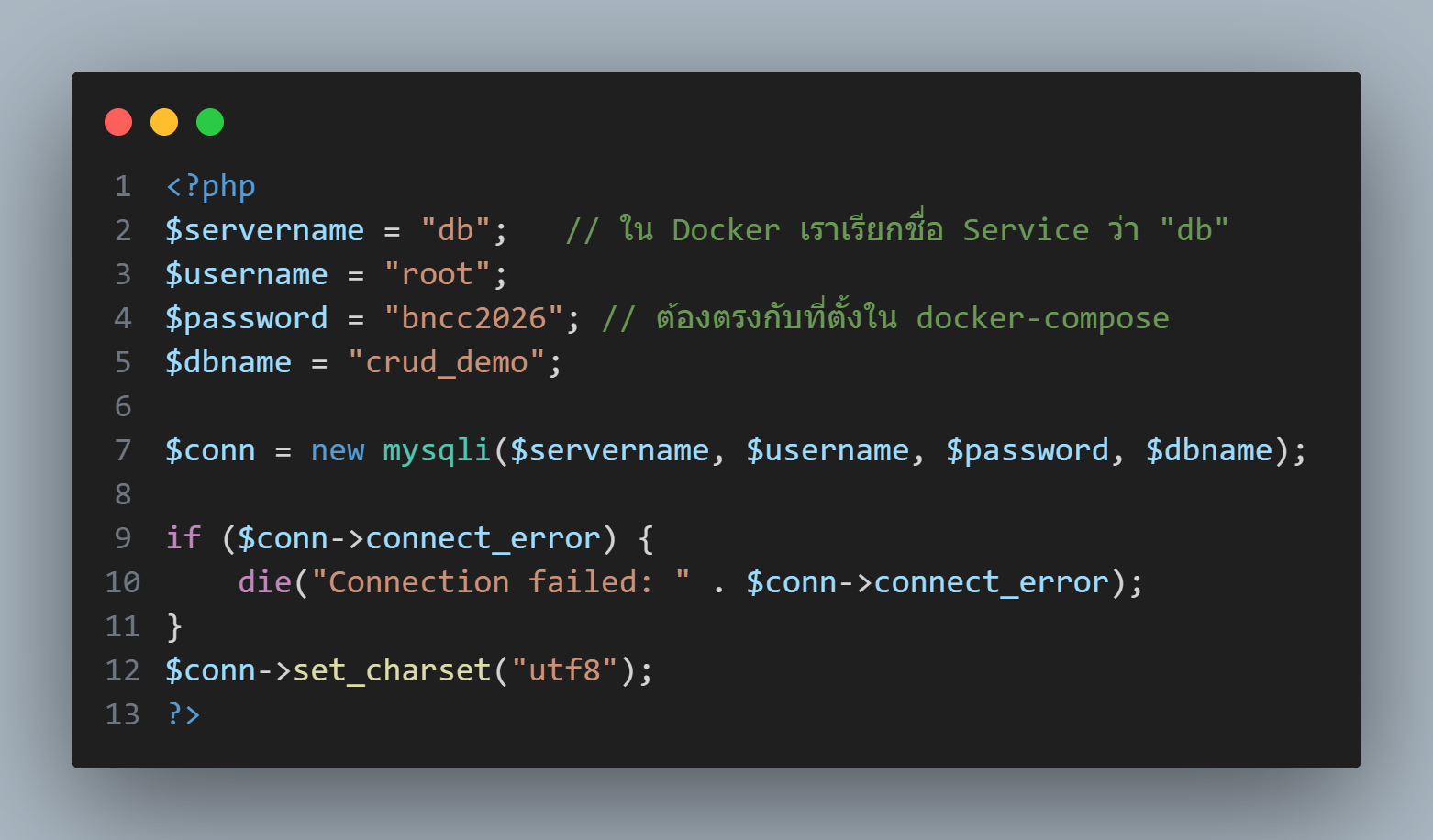
**รายละเอียดการแก้ไข:**

* Service Web: สร้าง Web Server โดย Map Port 80:80 เพื่อให้เข้าถึงผ่าน Browser ได้โดยตรง
* Service DB: สร้าง Database Server โดยใช้ Image MariaDB 10.6 และกำหนดรหัสผ่าน Root เป็น bncc2026
* Service PMA: สร้างระบบจัดการฐานข้อมูล (phpMyAdmin) โดยกำหนดให้เข้าใช้งานผ่าน Port 81 (81:80)
* Network: สร้างเครือข่ายจำลองชื่อ dev\_net (Bridge Driver) และกำหนดให้ทุก Service เชื่อมต่อผ่านเครือข่ายนี้ เพื่อให้สามารถสื่อสารกันได้ภายใน

**ส่วนที่ 2: การเชื่อมต่อฐานข้อมูล (Database Connection)**

**3. ไฟล์ db.php**

* **หน้าที่หลัก:** เป็นไฟล์กลางสำหรับสร้างการเชื่อมต่อระหว่าง PHP Application กับฐานข้อมูล
* **รายละเอียดการแก้ไข:**
  + เปลี่ยนค่าตัวแปร **$servername** จากเดิมที่มักใช้ **localhost** ให้กลายเป็น **db** **(ซึ่งเป็นชื่อ Service Name ที่ระบุไว้ใน docker-compose.yml)** เพื่อให้ **Docker Resolve IP Address** หากันเจอภายใน **Network dev\_net**
  + กำหนดค่า **$username** เป็น **root** และ **$password** เป็น **bncc2026** ให้ตรงกับการตั้งค่า **Environment Variable** ของ **Database Container**
  + กำหนดการเข้ารหัสภาษาเป็น **utf8** เพื่อให้รองรับการเก็บและแสดงผลข้อมูลภาษาไทยได้อย่างถูกต้อง

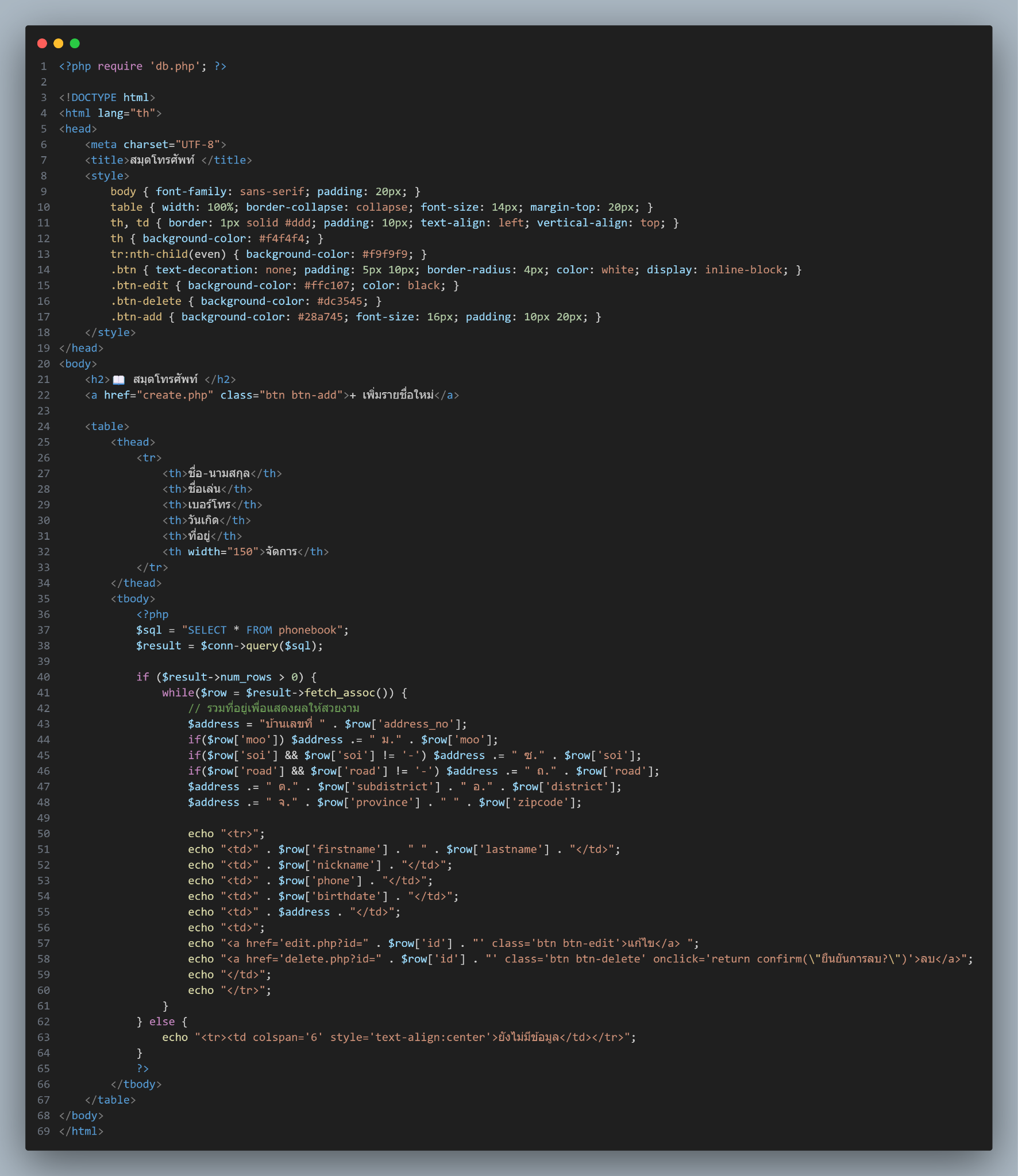


**ส่วนที่ 3: การพัฒนาโปรแกรมสมุดโทรศัพท์ (Phonebook Application)**

**4. ไฟล์ index.php (ส่วนแสดงผลข้อมูล - Read)**

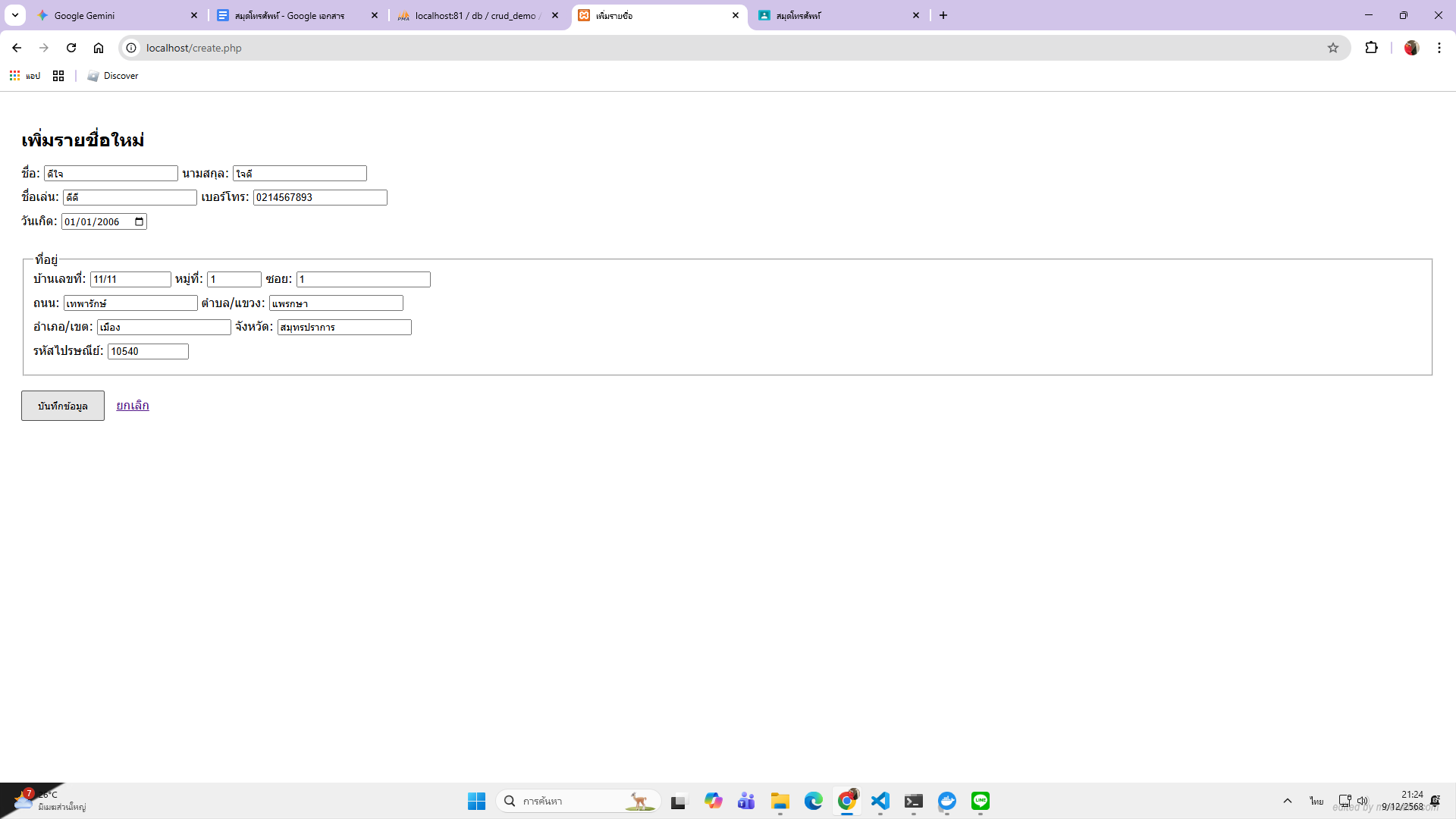
* **หน้าที่หลัก:** ดึงข้อมูลรายชื่อทั้งหมดจากตาราง phonebook มาแสดงผลในรูปแบบตาราง HTML
* **รายละเอียดการแก้ไข:**
  + แก้ไขคำสั่ง **SQL** เป็น **SELECT \* FROM phonebook(ชื่อตารางใหม่)**
  + ปรับปรุงหัวตาราง **(<th>)** ให้รองรับโจทย์ใหม่ ได้แก่ **ชื่อ-นามสกุล, ชื่อเล่น, เบอร์โทร, วันเกิด, และที่อยู่**
  + เขียนโค้ด **PHP** เพิ่มเติมเพื่อทำการรวมข้อมูลที่อยู่ (Concatenation) ซึ่งประกอบด้วย บ้านเลขที่, หมู่, ซอย, ถนน, ตำบล, อำเภอ, จังหวัด และรหัสไปรษณีย์ ให้แสดงผลรวมอยู่ในคอลัมน์เดียว เพื่อความสวยงามและอ่านง่าย
  + เพิ่มปุ่มสำหรับเชื่อมโยงไปยังหน้าแก้ไข **(edit.php)** และหน้าลบข้อมูล **(delete.php)** โดยส่งค่า id ไปด้วย



****

**5. ไฟล์ create.php (ส่วนเพิ่มข้อมูล - Create)**

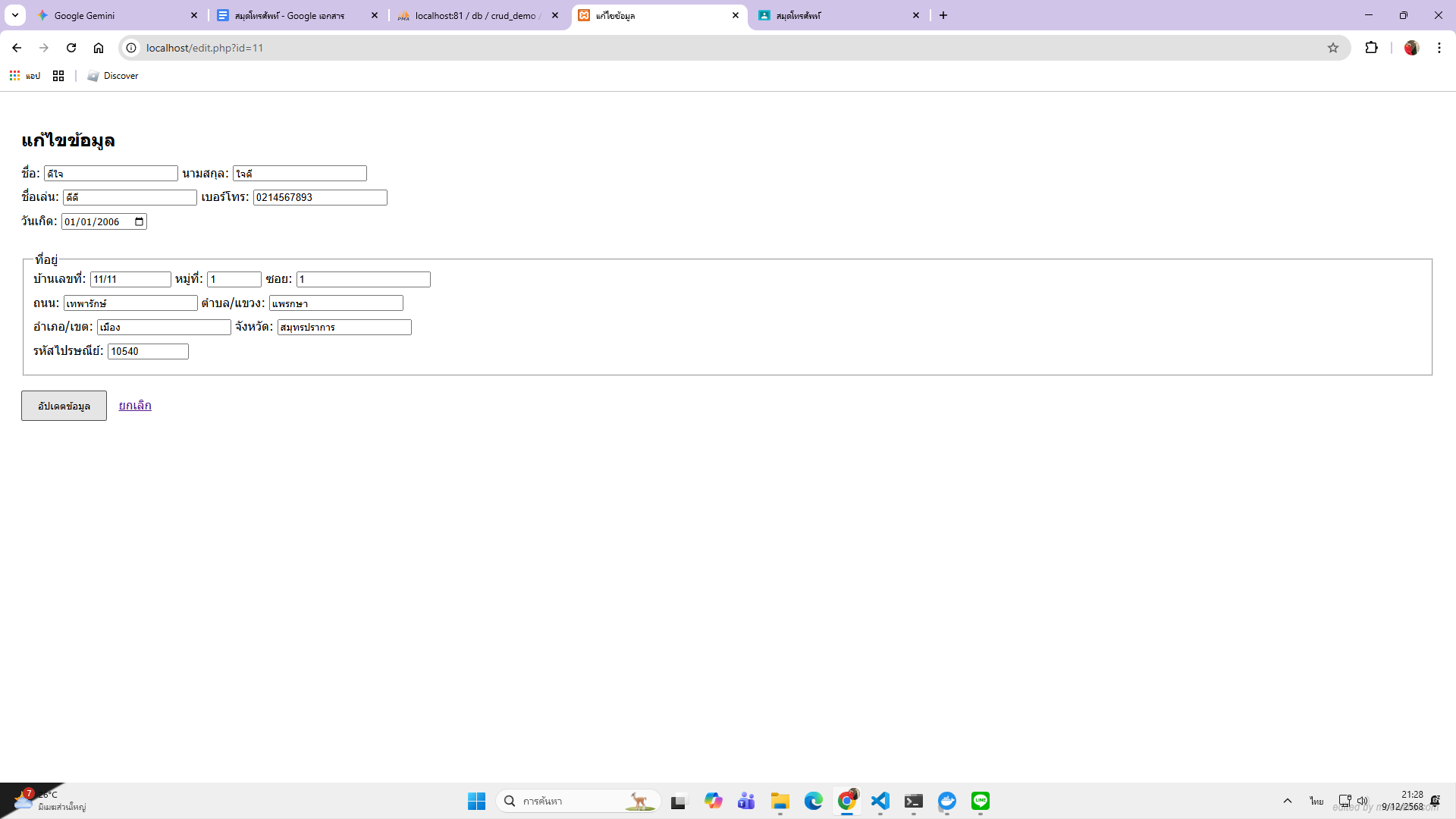
* **หน้าที่หลัก:** แสดงฟอร์มรับข้อมูลและบันทึกข้อมูลสมาชิกใหม่ลงฐานข้อมูล
* **รายละเอียดการแก้ไข:**
  + **ออกแบบฟอร์ม HTML ใหม่ ให้มีช่องกรอกข้อมูล (<input>) ครบทั้ง 13 รายการ เพื่อรองรับรายละเอียดที่อยู่ที่ซับซ้อนและข้อมูลส่วนตัว (ชื่อเล่น, วันเกิด, เบอร์โทร)**
  + **แก้ไขคำสั่ง SQL INSERT INTO ให้ระบุชื่อคอลัมน์ครบทั้ง 13 คอลัมน์**
  + **ใช้ Prepared Statement และแก้ไขฟังก์ชัน bind\_param ให้รองรับตัวแปร 13 ตัว ("sssssssssssss") เพื่อความปลอดภัยและป้องกัน SQL Injection**



****

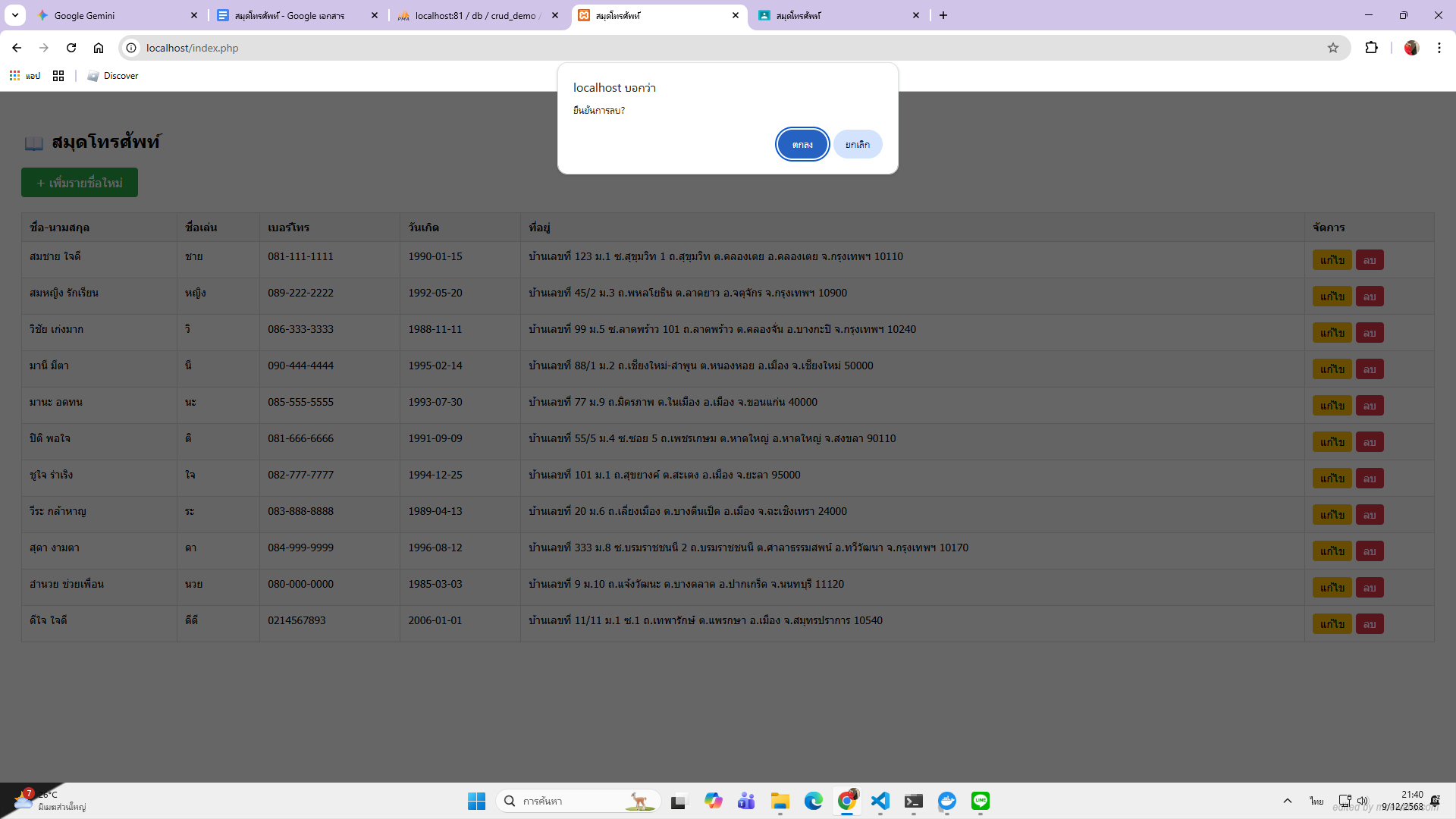
**6. ไฟล์ edit.php (ส่วนแก้ไขข้อมูล - Update)**

* **หน้าที่หลัก:** ใช้สำหรับปรับปรุงข้อมูลเดิมที่มีอยู่ในระบบ
* **รายละเอียดการแก้ไข:**
  + **ส่วนดึงข้อมูลเก่า: เขียนคำสั่ง SQL SELECT โดยอ้างอิงจาก id ที่ส่งมา เพื่อนำข้อมูลเดิมทั้ง 13 รายการ มาแสดงผลใน Attribute value ของ Input แต่ละช่อง เพื่อให้ผู้ใช้เห็นข้อมูลเดิมก่อนแก้ไข**
  + **ส่วนบันทึกข้อมูล: แก้ไขคำสั่ง SQL UPDATE ให้ทำการเปลี่ยนแปลงข้อมูลได้ครบทุกฟิลด์ (รวมถึงที่อยู่และข้อมูลส่วนตัวใหม่) โดยใช้เงื่อนไข WHERE id=? เพื่อให้แก้ไขถูกคน**



****

**7. ไฟล์ delete.php (ส่วนลบข้อมูล - Delete)**

* **หน้าที่หลัก:** ลบข้อมูลรายชื่อออกจากฐานข้อมูลถาวร
* **รายละเอียดการแก้ไข:**
  + **รับค่า id ที่ส่งมาจากหน้า index.php ผ่าน method GET**
  + **แก้ไขคำสั่ง SQL เป็น DELETE FROM phonebook WHERE id=? เพื่อลบข้อมูลออกจากตาราง phonebook (แทนตาราง users ) โดยใช้ Prepared Statement เพื่อความปลอดภัยในการสั่งลบข้อมูล**

****

### **ส่วนที่ 4 : การออกแบบฐานข้อมูล (Database Design)**

**1. การปรับปรุงโครงสร้างตาราง (Schema Design)**

* **รายละเอียด:**
  + **ได้ทำการสร้างตารางใหม่ชื่อว่า phonebook (แทนที่ตาราง users เดิม)**
  + **กำหนด Primary Key คือ id (Auto Increment) เพื่อใช้เป็นรหัสอ้างอิงที่ไม่ซ้ำกันในแต่ละรายชื่อ**

**2. การเพิ่มฟิลด์ข้อมูล (Data Fields) เพื่อให้ระบบสามารถจัดเก็บข้อมูลได้ครบถ้วนตามความต้องการ จึงได้กำหนดคอลัมน์ (Columns) ในตารางดังนี้:**

* **ข้อมูลส่วนตัว:**
  + **firstname, lastname: เก็บชื่อและนามสกุล**
  + **nickname: เก็บชื่อเล่น (เพิ่มใหม่)**
  + **phone: เก็บเบอร์โทรศัพท์ (เพิ่มใหม่)**
  + **birthdate: เก็บวันเดือนปีเกิด รูปแบบ DATE (เพิ่มใหม่)**
* **ข้อมูลที่อยู่ (เก็บแยกส่วนเพื่อความละเอียด):**
  + **ได้ทำการออกแบบให้แยกเก็บส่วนประกอบของที่อยู่ออกเป็นฟิลด์ย่อย ได้แก่ address\_no (บ้านเลขที่), moo (หมู่), soi (ซอย), road (ถนน), subdistrict (ตำบล/แขวง), district (อำเภอ/เขต), province (จังหวัด), และ zipcode (รหัสไปรษณีย์)**
  + **เหตุผล: การแยกเก็บข้อมูลที่อยู่จะช่วยให้การจัดการข้อมูลมีความยืดหยุ่น และสามารถนำมาแสดงผลต่อกัน (Concatenate) ได้อย่างถูกต้องในหน้าเว็บไซต์**

**3. คำสั่ง SQL ที่ใช้ (SQL Statement)**

**CREATE TABLE `phonebook` (**

**`id` int(11) NOT NULL AUTO\_INCREMENT,**

**`firstname` varchar(100) NOT NULL,**

**`lastname` varchar(100) NOT NULL,**

**`nickname` varchar(50) NOT NULL,**

**`phone` varchar(20) NOT NULL,**

**`birthdate` date NOT NULL,**

**`address\_no` varchar(50) NOT NULL,**

**`moo` varchar(20) DEFAULT NULL,**

**`soi` varchar(50) DEFAULT NULL,**

**`road` varchar(100) DEFAULT NULL,**

**`subdistrict` varchar(100) NOT NULL,**

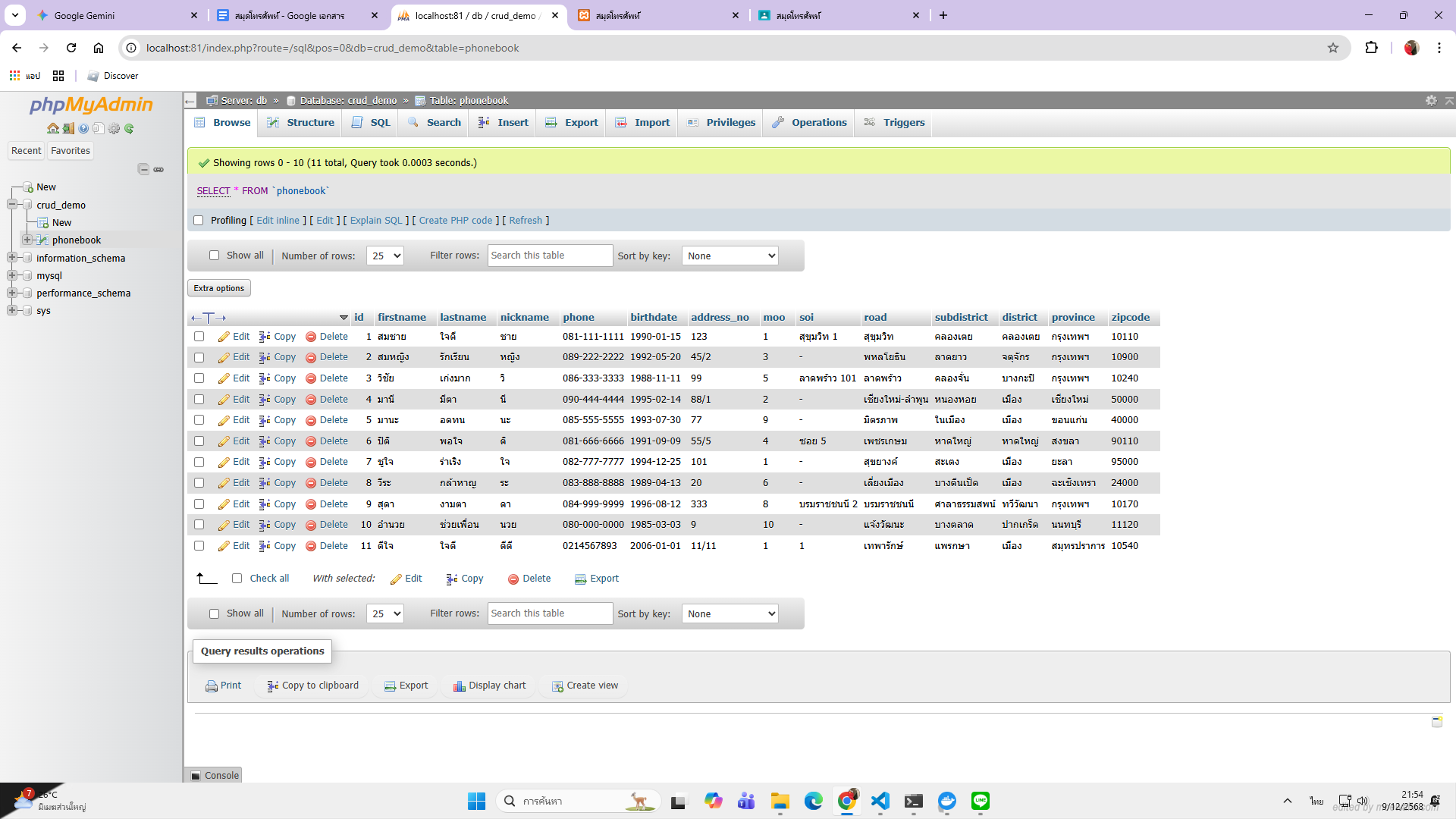
**`district` varchar(100) NOT NULL,**

**`province` varchar(100) NOT NULL,**

**`zipcode` varchar(10) NOT NULL,**

**PRIMARY KEY (`id`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**



**นาย อลงกรณ์ หงษ์คะนาน เลขที่ 17**