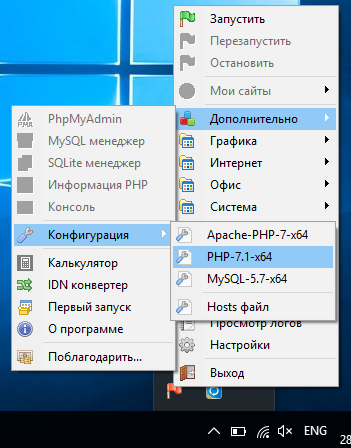
Лабораторна робота №6.

**SQLite. Написання клієнтського додатку на мові PHP.**

1. Під час виконання даної лабораторної роботи будемо використовувати пакет OpenServer, що включає також скриптову мову програмування PHP. OpenServer не вимагає інсталяції, папка з ПО може бути скопійована на диск, після чого потрібно просто запустити Open Server x64.exe або Open Server x86.exe в залежності від версії ОС.
2. В PHP підтримка SQLite починаючи із версії 5 включена автоматично, тому ніяких додаткових розширень встановлювати не доведеться. Переконаємось лише, що відповідний модуль доступу до даних підключено в розділі extensions нашого php.ini.

За умови використання OpenServer у ролі веб-серверу, запускаємо «Дополнительно» -> «Конфигурация» -> «PHP \*».



Та у вікні текстового редактора, що з’явився, переконуємося, що рядок «extension=php\_pdo\_sqlite.dll» розкоментовано (у разі необхідності прибираємо перший символ рядку «;» та зберігаємо файл.



Якщо веб-сервер було запущено, він має бути перезавантажений.

1. Далі для роботи із БД будемо використовувати вбудований клас SQLite3. При створенні екземпляра класу відувається автоматичне підключення БД, отже в конструктор необхідно передати параметр – ім’я файлу БД.

$myDB = new SQLite3(‘myDatabase.db’);

Крім цього, будемо використовувати ще 2 методи класу SQLite3.

Перший - метод exec(), наприклад

$myDB->exec(‘create table t1(f1 int);’);

якому у якості параметру передається SQL запит для виконання. Метод повертає істину у разі успішного виконання, і хибність у випадку виникнення помилки.

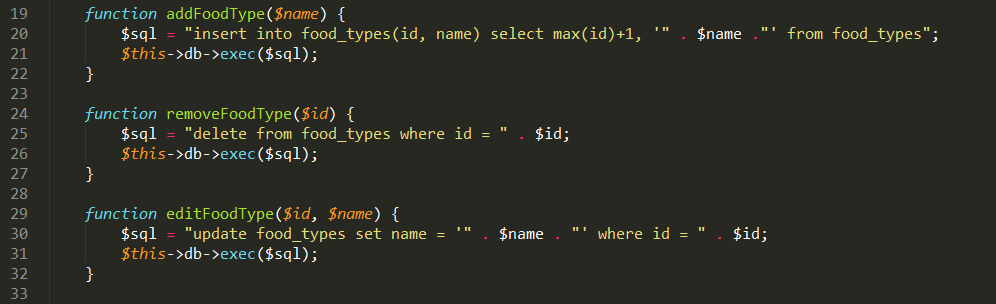
Другий – метод query(), він є дещо схожим на попередній, також приймає параметр - SQL запит для виконання але повертає об’єкт SQLite3Result, або false у випадку виникнення помилки.

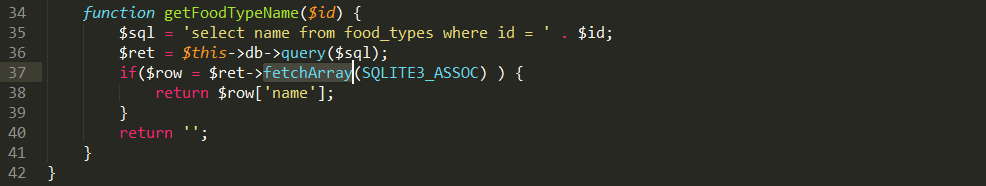
Клас SQLite3Result представляє доступ до результуючого набору розширення SQLite3. В цьому класі використаємо метод fetchArray, що повертає рядок із результуючого набору. Метод викликається із параметром SQLITE3\_ASSOC, що повертає асоціативний масив, у якому індекс відповідає імені стовпчика у результуючому наборі.

1. Переходимо до реалізації програмного коду, що відповідатиме за відображення, додавання, зміну та видалення даних із таблиці «Типи страв» бази даних «Їжа». В папці з OpenServer у директорії domains/localhost/ папку «sqlite». В цій папці створимо файл sqlite.php, в якому створимо клас Food, що буде відповідати за операції із БД. Підключатися до БД будемо під час створення екземпляра, тобто у конструкторі. Крім того знадобиться ще 5 методів для отримання списку типів їжі, читання одного з типів їжі, додавання, зміни та видалення типи їжі відповідно.

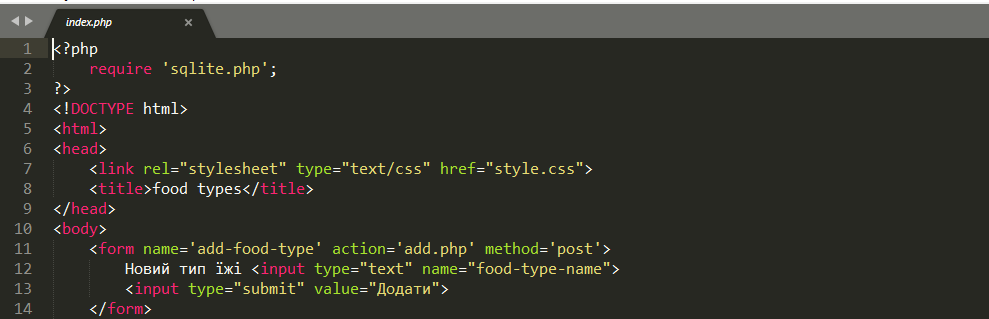
Нижче наведено клас Food

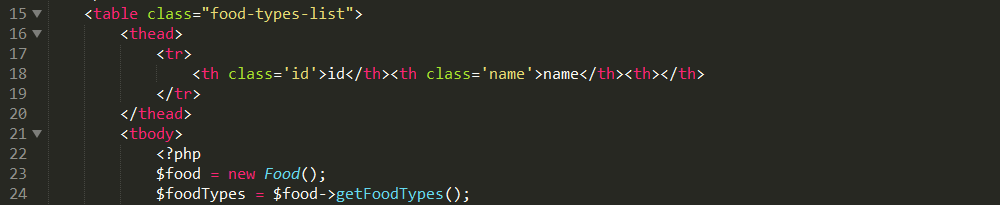


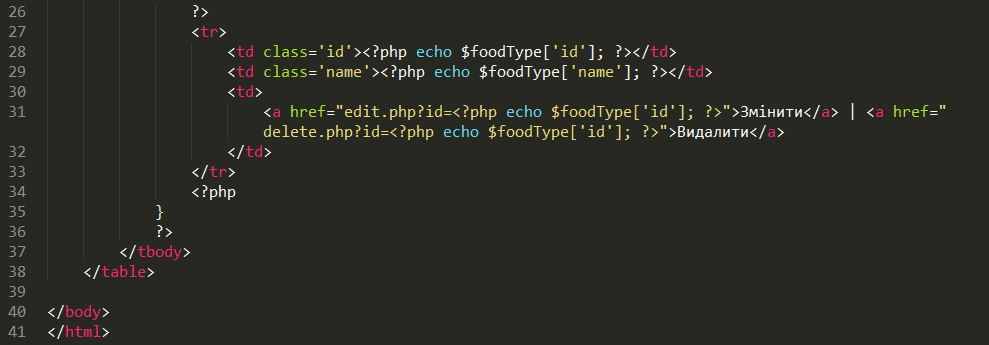




1. Далі створимо index.php, що буде відповідати за відображення списку типів їжі у таблиці. Крім того, тут реалізуємо форму додавання нового типу їжі. На початку підключаємо sqlite.php та далі у коді створюємо екземпляр Food та використовуємо метод getFoodTypes() для отримання вмісту таблички «Типи їжі».





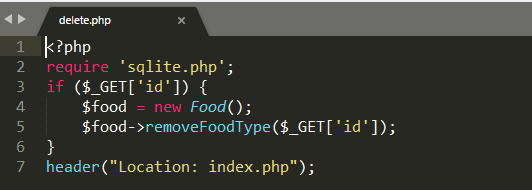


1. В головному файлі ми посилаємось на add.php, що має додавати новий рядок до таблиці, delete.php для видалення рядку, та edit.php, в якому реалізована форма редагування та безпосередньо сама зміна даних у БД. Наведемо програмний код їх реалізації.

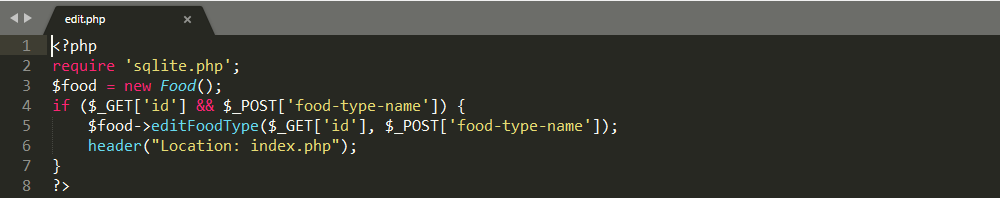
Add.php

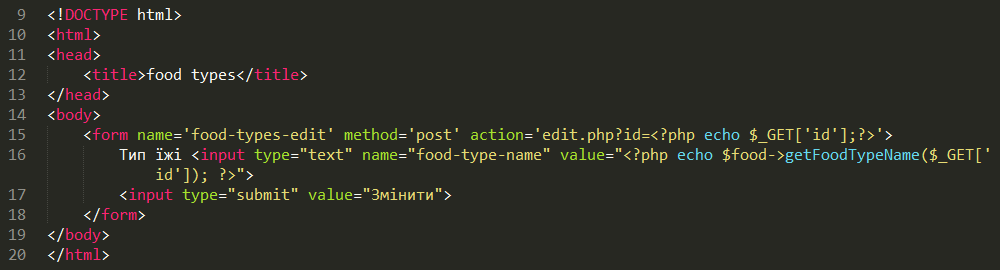


Delete.php

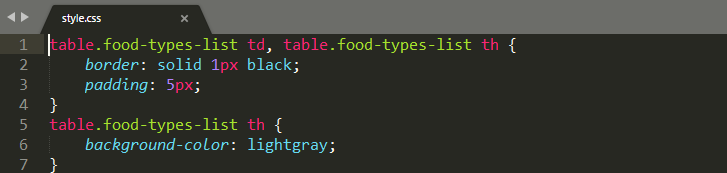


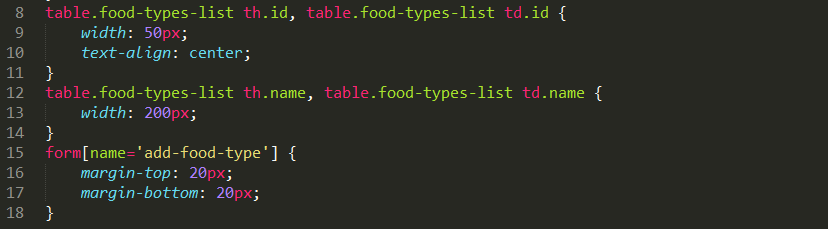
Edit.php



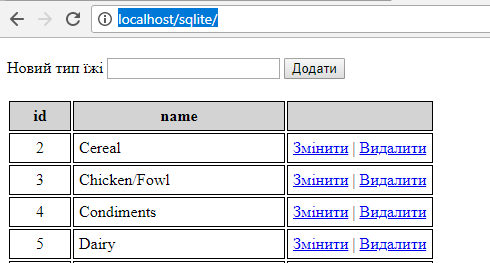


1. Для форматування виводу створимо style.css та підключимо його в index.php





1. Перевіримо роботу додатку, набравши в адресному рядку браузеру «http://localhost/sqlite/». Спробуємо додати, змінити та видалити дані.



1. Кінець роботи