



Національний університет  
водного господарства  
та природокористування

Міністерство освіти і науки України  
Національний університет водного господарства  
та природокористування  
Кафедра геодезії та картографії

05-04-26

**МЕТОДИЧНІ ВКАЗІВКИ**  
до виконання лабораторних робіт з дисципліни  
**«Побудова та управління банками геоінформації»**  
студентами напряму підготовки 6.080101 „Геодезія, картографія та  
землеустрій” професійного спрямування „Геоінформаційні системи  
і технології”  
Частина 3 «Робота з MySQL та PHP»

Рекомендовано до друку  
методичною комісією напряму  
підготовки 6.080101 „Геодезія,  
картографія та землеустрій”  
Протокол № 5 від 02.03. 2015 р.

м. Рівне – 2015 р.



Національний університет

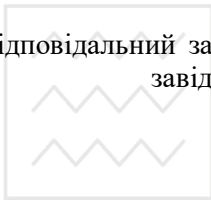
водного господарства

та природокористування

Методичні вказівки до виконання лабораторних робіт з дисципліни «Побудова та управління банками геоінформації» студентами напряму підготовки 6.080101 „Геодезія, картографія та землеустрій” професійного спрямування „Геоінформаційні системи і технології” Частина 3 «Робота з MySQL та PHP» / О. Є. Янчук, А. В. Прокопчук, Н. В. Левчук. – Рівне: НУВГП, 2015. – 36 с.

Упорядники: О. Є. Янчук, канд. техн. наук, доцент кафедри геодезії та картографії;  
А. В. Прокопчук, асистент кафедри геодезії та картографії  
Н. В. Левчук, асистент кафедри геодезії та картографії

Відповідальний за випуск: Р. М. Янчук, канд. техн. наук, доцент, завідувач кафедри геодезії та картографії.



Національний університет  
водного господарства  
та природокористування

© Янчук О. Є., Прокопчук А. В.,  
Левчук Н. В., 2015  
© НУВГП, 2015



## ЗМІСТ

ВСТУП.....	4
ЛАБОРАТОРНА РОБОТА № 1. ВИБІР ТА РЕЄСТРАЦІЯ НА WEB-ХОСТИНГУ .....	5
1. Що таке хостинг? .....	5
2. Класифікація хостингів .....	5
3. Основні вимоги до майбутнього хостингу .....	7
4. Реєстрація та управління веб-хостингом .....	9
ЛАБОРАТОРНА РОБОТА № 2. ЗНАЙОМСТВО З СИСТЕМОЮ КЕРУВАННЯ БАЗАМИ ДАНИХ MYSQL .....	10
1. Огляд MySQL та phpMyAdmin.....	10
2. Синтаксис основних команд на мові SQL .....	11
ЛАБОРАТОРНА РОБОТА № 3. АДМІНІСТРУВАННЯ БАЗИ ДАНИХ ЗА ДОПОМОГОЮ PHPMYADMIN.....	13
1. Створення бази даних .....	13
2. Створення таблиць та налаштування полів.....	15
3. Основні операції з даними в phpMyAdmin.....	17
ЛАБОРАТОРНА РОБОТА № 4. ЗНАЙОМСТВО З МОВОЮ PHP. 18	
1. Опис мови програмування PHP .....	19
2. Особливості мови PHP.....	19
3. Робота з рядками .....	20
4. Змішані типи даних .....	21
5. Конструкції мови .....	22
6. Базовий синтаксис .....	23
ЛАБОРАТОРНА РОБОТА № 5. СТВОРЕННЯ ІНТЕРАКТИВНИХ ФОРМ ДЛЯ ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ.....	25
1. Створення інтерактивної форми для введення даних..	25
2. Реалізація виведення даних із бази.....	27
РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА РЕСУРСИ.....	29
ДОДАТОК. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК.....	30



## ВСТУП

---

Сучасні інформаційні технології – це методи і засоби для збору, зберігання, обробки та отримання інформації на основі сучасних засобів обчислювальної техніки. Складовими частинами будь-якої інформаційної системи є бази даних (БД) і додатки для обробки даних.

Головним завданням курсу є навчання студентів навичкам проектування, організації функціонування і використання банків даних геоінформаційних систем, а також впровадження таких систем в виробництво і органи державного управління та у всі можливі галузі народного господарства.

У даній методичній розробці викладені основні положення, порядок виконання та розробки бази даних за допомогою системи керування базами даних MySQL та засобу адміністрування цих баз phpMyAdmin, також тут подані основи скриптової мови програмування PHP та порядок реалізації інтерактивних форм для введення та виведення інформації із віддалених баз даних.

У результаті виконання даних лабораторних робіт студенти повинні навчитися реєструватись на хостингу, створювати та налаштовувати віддалені бази даних за допомогою MySQL (phpMyAdmin), створювати форми для додавання та виведення інформації, збереженої у віддалених базах даних, за допомогою мови PHP та програми Dreamweaver.



## ЛАБОРАТОРНА РОБОТА № 1

### ВИБІР ТА РЕЄСТРАЦІЯ НА WEB-ХОСТИНГУ

1. Що таке хостинг?
2. Класифікація хостингів.
3. Основні вимоги до майбутнього хостингу.
4. Реєстрація та управління веб-хостингом.

#### 1. ЩО ТАКЕ ХОСТИНГ?

**Хостинг** – послуга, що надає дисковий простір для розміщення фізичної інформації на сервері, що постійно перебуває в мережі (наприклад Internet).

Зазвичай, під поняттям послуги хостингу мають на увазі, як мінімум, послугу розміщення файлів сайту на сервері, на якому запущене програмне забезпечення, необхідне для обробки запитів до цих файлів (веб-сервер). Як правило, до послуг хостингу вже входить надання місця для поштової кореспонденції, баз даних, DNS файлового сховища тощо, а також підтримка функціонування відповідних сервісів, однак вони можуть надаватися і окремо. Розрізняють безкоштовний та платний хостинг. Безкоштовні «гостери» заробляють на тому, що розміщують рекламу на своїх сайтах.

#### 2. КЛАСИФІКАЦІЯ ХОСТИНГІВ

**Хостинг можна класифікувати залежно від:**

- місця розташування технічного майданчика;
- програмного забезпечення, що встановлено на сервері;
- типу сервера, на якому розміщуються файли сайтів або програми;
- умов надання;
- типу об'єктів, що розміщуються на серверах хостинг-провайдеру.

За **місцем розташування** хостинг може бути, наприклад: українським (технічний майданчик розташовано в Україні), російським (в Росії), американським (у США) тощо.

За **типом програмного забезпечення** хостинг поділяють на UNIX-хостинг і WINDOWS-хостинг.



- **RedHat LINUX (UNIX-платформа)** - традиційна платформа веб-хостингу. Це надійна і міцна платформа для підтримки сайтів з великим об'ємом інформації і великою кількістю звернень (трафіком);

- **Windows NT (WINDOWS-платформа)** забезпечує певні опції, які не доступні на платформі хостингу Unix, такі як ASP, Microsoft SQL 2000, Access databases і Cold Fusion 5. Тому, якщо для сайту потрібне одне або кілька з цих програмних застосувань, варто обирати WINDOWS-платформу.

За **типом сервера** на якому розміщуються файли сайтів або програм:

- **віртуальний сервер** – виділення місця на диску для розміщення веб-сайтів, середовище виконання веб-сервісів єдине для багатьох користувачів, ресурси розподілені між усіма користувачами на одному сервері, де може розміщуватись від 50 до 1000 користувачів. Невеликі дешеві хостинг-провайдери часто нехтують безпекою і взагалі не розмежовують привілеїв користувачів, що дозволяє одному користувачеві на сервері мати доступ до сайтів сотень інших користувачів);

- **віртуальний виділений сервер (VPS або VDS)** – послуга, в рамках якої користувачеві надається так званий віртуальний виділений сервер. Це повноцінна альтернатива фізичного виділеного сервера з великою кількістю переваг, високою стабільністю, простотою в управлінні і настройці, стійкістю до відмов і набагато меншими фінансовими витратами;

- **виділений сервер** – послуга з надання сервера цілком. Використовується для реалізації нестандартних завдань (сервісів), а також розміщення «важких» веб-проектів, які не можуть співіснувати на одному сервері з іншими проектами і вимагають під себе всі ресурси сервера;

- **колокація (Collocation)** – надання місця в дата-центрі провайдера для обладнання клієнта (зазвичай шляхом монтажу в стійці) і підключення його до мережі Internet.

За **умовами надання** хостинг поділяється на:

- платний;
- безкоштовний.

Зазвичай компанія, що надає безкоштовний хостинг, заробляє шляхом показу реклами на сторінках, розміщених на ньому.

Безкоштовний хостинг, як правило, повільніше платного, надає тільки базові послуги і ненадійний. Також часто безкоштовний хостинг надається для того, щоб через якийсь час перевести користувача на платну основу, шляхом погіршення умов користування. Приватні особи для своїх домашніх сторінок на початковому етапі їх розвитку переважно використовують безкоштовний хостинг. Громадські організації можуть використовувати як платний хостинг, так і безкоштовний. Комерційні організації практично завжди користуються послугами платного хостингу.

За **типом об'єктів**, що розміщуються на серверах хостинг-провайдера:

- веб хостинги (розміщення сайтів);
- E-mail хостинги (розміщення електронних скриньок);
- DNS хостинги (розміщення доменних імен);
- тощо.

### 3. ОСНОВНІ ВИМОГИ ДО МАЙБУТНЬОГО ХОСТИНГУ

Вибір хост-провайдера є основою успішного представництва в мережі. Тому, варто знати, які вимоги слід висувати до провайдера хостингу.

1. Багато про провайдера свідчить його сайт – візитна картка компанії. Хост-провайдери мають надавати на своїх сайтах повну і однозначну інформацію про себе і послуги, щоб у потенційного клієнта не виникало додаткових питань. Якщо на сайті немає інформації з основних питань – це показник недбалого відношення до послуг, що надаються.

2. Хостинг компанії мають бути професіоналами не лише в наданні послуг, але і у відносинах з клієнтами. Тому, слід перевірити чи існує служба технічної підтримки і чи задовольняє вона всім можливим вимогам.

3. Потужні хостинг-компанії надають своїм клієнтам можливість тестування зручності і якості наявних сервісів за допомогою демо-входу в панель управління акаунтом (особистої сторінки користувача). Слід перевірити швидкість доступу до сайтів, які розміщені у даного провайдера в різні години доби і дні тижня, щоб переконатися в якості запропонованих послуг. Слід



дізнатися про процедуру повернення грошей, якщо якість послуг провайдера виявиться незадовільною.

4. Чи надає хостинг у розпорядження піддомени? В середньому, середньостатистичний хост-провайдер надає близько 6 піддоменів у розпорядження користувача, не стягуючи додаткової плати. Слід з'ясувати умови користування піддоменами. Піддомени можуть виявитися корисним доповненням до сайту користувача. Наприклад, якщо на сайті відкривається форум, то можна дати йому зручну адресу: <http://forum.site.ua/>.

5. Перевірити швидкість передачі даних. Бажано відвідати сайти, які розташовані на серверах компанії і прослідкувати час завантаження сторінок. Також, варто дізнатися враження клієнтів і замовників, що користуються послугами хостингу в даній компанії.

6. Дізнатися про систему захисту серверів, які використовує компанія.

7. Якщо користувач має свій домен, провайдер зобов'язаний надати у його розпорядження значну кількість поштових імен у межах його домену (*name@site.ua*).

8. Більшість хост-провайдерів надають FTP-доступ та доступ до бази даних. Обов'язково з'ясувати у провайдера, чи надає він наступні сервіси: CGI-сервіси, підтримку Perl, C, PHP MYSQL? Чи не стягується за використання цих послуг додаткова плата?

9. Вартість хостингу також грає важливу роль. Наприклад, ціна за оренду виділеного сервера може досягати порядку 200 \$ у місяць, тоді як плата за звичайний хостинг для сайту – в місяць може складати всього 2-3 \$.

#### **Основні критерії вибору хостингу:**

- вартість хостингу;
- сервісні послуги;
- параметри хостингу;
- якість послуг, що надаються.

Очевидно, що ці критерії є важливими при виборі хостингу: співвідношення ціни, наявність сервісних послуг зумовлюють вибір користувача на користь того або іншого хостинг-провайдера, що гарантує надійну роботу сайту та його окупність.

Отже, хостинг це не лише оренда місця на твердому диску сервера і під'єднання до швидкісних каналів інформації. Сучасні сайти мають різний ступінь складності: від домашніх сторінок до





великих порталів. Для їх нормального відображення і безперебійної роботи необхідно мати цілий ряд додаткових функцій, які повинен виконувати хост-сервер. Ці функції реалізовано в багатьох хостинг-провайдерів шляхом встановлення додаткових допоміжних програм на серверах.

Дізнатися про хостинг-провайдерів просто, для цього достатньо задати в пошуковій системі запит типу «хостинг» або «купити хостинг», або «хостинг для сайту» і зазвичай, на перші позиції буде виведено сайти відомих, потужних та надійних хостинг-провайдерів.

Як варіант, можна обрати один із таких безкоштовних хотингів:

<http://host1free.com/>

<http://o0.pm/uk/>

<http://hostinger.com.ua/>

#### 4. РЕЄСТРАЦІЯ ТА УПРАВЛІННЯ ВЕБ-ХОСТИНГОМ

Спробуємо зареєструватись на хостингу <http://hostinger.com.ua/>.

Перейшовши на сайт провайдера натискаємо на посилання “Реєстрація” в правому верхньому куті. Далі, задаємо реєстраційні дані та підтверджуємо свою електронну скриньку перейшовши за посиланням у листі відправленому з хостингу.

Заходимо у свій кабінет ввівши e-mail та пароль. Обираємо “Хостинг” -> “Новий аккаунт” -> “Замовити (Тарифний план безкоштовний)”. Відкриється діалогове вікно у якому потрібно ввести (обрати): тип домену (субдомен); ім'я домену; пароль. По завершенню заповнення форми натискаємо “Продовжити”.

Після створення аккаунту переходимо на панель управління цим аккаунтом (“Хостинг” -> ”Ваш аккаунт”).

Відкривається вікно панелі керування у якому всі інструменти розподілені за наступними розділами: “Аккаунты” (перегляд загальної інформації та зміна тарифного плану), “Почта” (містить інструменти для керування поштовими аккаунтами), “Домены” (містить інструменти для управління доменними іменами, в тому числі і для реєстрації (підключення) нових доменів), “Сайт” (містить інструменти для створення та імпорту сайтів), “Файлы” (містить інструменти для роботи із файлами та їх резервного копіювання), “Базы данных” (містить інструменти для створення, копіювання,



## ЛАБОРАТОРНА РОБОТА № 2

### ЗНАЙОМСТВО З СИСТЕМОЮ КЕРУВАННЯ БАЗАМИ ДАНИХ MYSQL

---

1. Огляд MySQL та phpMyAdmin.
2. Синтаксис основних команд на мові SQL.

#### 1. ОГЛЯД MYSQL ТА PHPMYADMIN

**MySQL** – вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL – компактний багатоканальний сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх web-додатків. Початкові коди сервера компілюються на багатьох платформах. Найкраще можливості сервера виявляються в UNIX-системах, де є підтримка багатоканальності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Одним з найкращих інструментів для адміністрування СКБД MySQL є phpMyAdmin.

**PhpMyAdmin** – веб-застосунок з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування СКБД MySQL. PhpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати СКБД MySQL без безпосереднього вводу SQL команд, через дружній інтерфейс і з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення.

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей застосунок як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

## 2. СИНТАКСИС ОСНОВНИХ КОМАНД НА МОВІ SQL

До основних операцій, які виконуються під час роботи із таблицями БД можна віднести такі:

- INSERT – вставка рядка;
- SELECT – вибірка;
- UPDATE – зміна;
- DELETE – видалення.

**INSERT** – оператор мови SQL, котрий додає рядки в таблицю або переносить значення з однієї частини таблиці в іншу. В реляційній СКБД можна визначити два варіанти оператора INSERT:

- *однорядковий* оператор INSERT дозволяє додавати в таблицю один новий рядок (широко використовується в повсякденних аплікаціях, наприклад, програмах введення даних);
- *багаторядковий* оператор INSERT забезпечує витягування даних з однієї частини бази даних, їх трансформацію і додавання в іншу частину (використовується зазвичай при пакетній обробці і створенні нових даних).

Синтаксис оператора INSERT виглядає так:

**INSERT INTO** <Назва таблиці> (колонка1, [колонка2, ... ]) **VALUES** (значення1, [значення2, ...]) (однорядковий);

**INSERT INTO** <Назва таблиці> (колонка1, [колонка2, ... ])  
**SELECT** (колонка1а, [колонка2а, ...]) **FROM** <Назва таблиці  
 вибірки> **WHERE**... (дворядковий).

Приклад:

**INSERT INTO Students (ID, Name, Address) VALUES ('1', 'Sergiy  
 Sergiyenko', 'vul.Studentska, 6/400');**

**SELECT** – оператор мови SQL, котрий використовується для вибору даних з однієї чи багатьох таблиць. Повний синтаксис оператора **SELECT** є складним, проте його можна описати так:

**SELECT** <Список\_вибірки>  
 [ **INTO** <Назва нової таблиці> ]  
**FROM** <Назва таблиці>  
 [ **WHERE** <Умови\_пошуку> ]  
 [ **GROUP BY** <Поле для групування> ]  
 [ **HAVING** <Умови пошуку> ]  
 [ **ORDER BY** <Сортувати за умовою [ *ASC* | *DESC* ]> ]

Приклади:

**SELECT \* FROM T;** (в результаті будуть вибрані всі елементи всіх рядків таблиці T).

**SELECT C1 FROM T;**(в результаті буде вибрано стовпчик C1 та всі рядки в ньому).

**SELECT \* FROM T WHERE C1 = 1;** (будуть вибрані всі елементи всіх рядків де стовпчик C1 має значення 1).

**SELECT \* FROM T ORDER BY C1 DESC;** (в результаті буде вибрано ті ж самі рядки що й у першому запиті, але результат буде відсортовано в зворотному порядку (Z-A), тому що було використане ключове слово **DESC** оператора **ORDER BY** з полем C1 в якості сортування).

**UPDATE** – оператор мови SQL, що дозволяє оновити значення в заданих стовпцях таблиці.

Синтаксис оператора **UPDATE** виглядає так:

**UPDATE** <Назва таблиці> **SET** (<Ім'я поля> = <Значення>,  
 [<Ім'я поля 2> = <Значення>,...]) [ **WHERE** <Критерій відбору>];

Після виконання оператора **UPDATE** для всіх записів, які відповідають заданому критерію відбору, змінюється значення полів.



Приклад:

**UPDATE Students SET Adressa = 'vul.Studentska, 8' WHERE Id='1'** (в результаті буде змінено значення в полі Adressa у рядку де Id = 1).

**DELETE** – оператор видалення записів з таблиці. Критерій відбору записів для видалення визначається виразом WHERE. У разі, якщо критерій відбору не визначений, виконується видалення всіх записів.

Синтаксис оператора DELETE:

**DELETE FROM <Назва таблиці> WHERE <Умова відбору записів>;**

Наслідком виконання такої команди буде видалення тих рядків з таблиці <Назва таблиці>, які відповідають умові <Умова відбору записів>. При цьому ніякого результату команда не повертає і, отже, не може бути використана в якості параметра в команді SELECT.

Приклади:

**DELETE FROM Students WHERE Id='1'** (буде видалено рядок в таблиці Students, у якому Id=1).

**DELETE FROM Students WHERE \*** (буде видалено всі записи із таблиці Students).

## ЛАБОРАТОРНА РОБОТА № 3 АДМІНІСТРУВАННЯ БАЗИ ДАНИХ ЗА ДОПОМОГОЮ PHPMYADMIN

1. Створення бази даних.
2. Створення таблиць, налаштування полів.
3. Основні операції з даними в phpMyAdmin

### 1. СТВОРЕННЯ БАЗИ ДАНИХ

Під час роботи із сайтами (розробці, тестуванні модулів встановленні CMS, перегляді сторінок, тощо) часто виникає необхідність звертатися до бази даних.

Для створення бази даних будемо використовувати одну із найрозповсюдженіших систем адміністрування баз MySQL – phpMyAdmin.

Щоб мати можливість зайти в phpMyAdmin, він повинен бути встановлений на вашому локальному сервері або хостингу в інтернеті.

Щодо хостингу в інтернеті, то phpMyAdmin використовується практично всюди. У нашому випадку (хостинг [hostinger.com.ua](http://hostinger.com.ua)) на панелі керування хостингом достатньо обрати пункт MySQL, ввести ім'я бази та задати пароль, після цього натиснути на посиланні phpMyAdmin та обрати створену БД.

У іншому випадку, для запуску phpMyAdmin на локальному сервері необхідно в браузері перейти за певною адресою:

- Denwer – localhost/Tools/phpMyAdmin/;
- XAMPP – localhost/phpmyadmin/;
- WAMP – localhost/phpmyadmin/;
- LAMP – localhost/phpmyadmin/.

Практично у всіх середовищах розробки адреса та сама, окрім Denwer.

Отже, заходимо в phpMyAdmin і бачимо вікно, яке представлено на рис. 3.1. Воно поділене на 3 колонки (зліва направо): список ваших баз даних; робота з базами даних; дані про версії встановлених компонентів і додаткова службова інформація.

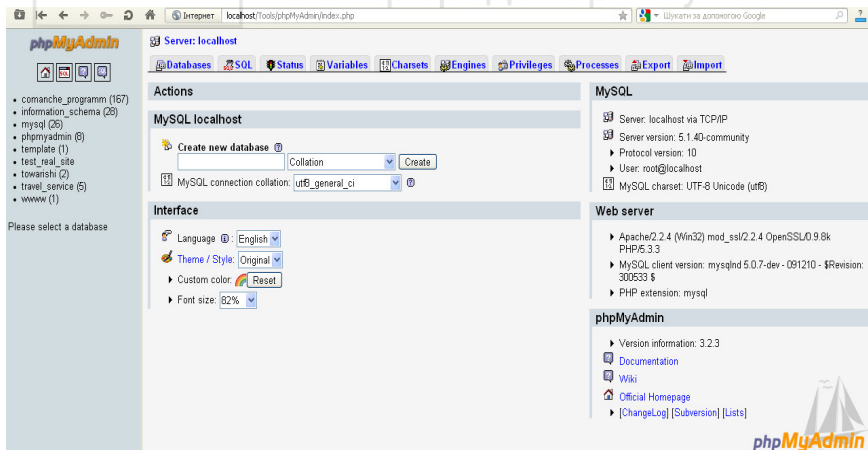


Рисунок 3.1 – Стартове вікно phpMyAdmin

Зверху розміщено горизонтальне меню з яким доведеться дуже часто працювати.



У секції **Interface** ми маємо змогу налаштувати інтерфейс програми: мову, стиль, колір і кегль тексту.

Слід зазначити, якщо ми працюємо із сервісом `hostinger.com.ua`, створення баз даних здійснюється у пункті MySQL, а не через phpMyAdmin.

Повернемось до створення бази даних. У блоці **MySQL Localhost** (рис. 3.2) задаємо параметри нової бази даних.

## MySQL localhost

**Create new database** ?

newsite utf8\_general\_ci Create

MySQL connection collation: utf8\_general\_ci ?

**Interface**

Language i: English

Theme / Style: Original

Custom color: Reset

Font size: 82%

Рисунок 3.2 – Діалог створення БД

В полі **Create new database** вписуємо назву бази (наприклад `newsite`), а у випадяючому списку обираємо кодування майбутньої БД. Далі натискаємо кнопку **Create**.

## 2. СТВОРЕННЯ ТАБЛИЦЬ ТА НАЛАШТУВАННЯ ПОЛІВ

Коли база створена відкриваємо її за допомогою phpMyAdmin (рис. 3.3).

Як бачимо база даних пуста. В першу чергу потрібно створити таблиці. Для цього у блоці **Create table** вводимо назву (наприклад “Students”) та кількість полів (напр. “3”) першої з них і натискаємо **Go**.

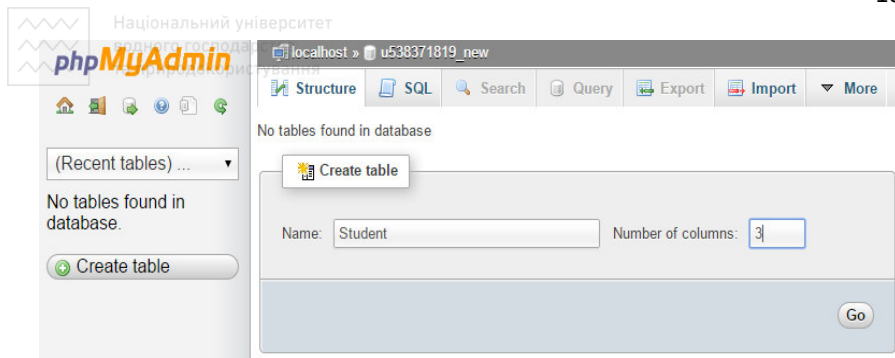


Рисунок 3.3 – Вікно управління БД

Запуститься вікно налаштування таблиці (рис. 3.4) в якому можна задати такі параметри:

**Name (Назва поля)** – заповнюємо латинськими літерами, цифрами та знаком підкреслювання, інші символи краще не використовувати.

**Type (Тип даних)** – для початку зупинимось на найбільш розповсюджених:

- цілі числа, TINYINT – приймає значення від -128 до 127, SMALLINT - приймає значення від -32768 до 32767, INT - приймає значення від -2 147 483 648 до +2 147 483 647.
- числа з плаваючою крапкою, FLOAT – мінімальне значення  $+/-1.175494351^{-39}$ , максимальне значення  $+/-3.402823466^{+38}$ .
- текстові поля, CHAR – рядок від 1 до 255 символів; VARCHAR – будь-якої заданої довжини; TEXT - тестове поле.

Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments	MIME type
id	INT	11	None			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
name	VARCHAR	100	None	utf8_unicode_c		<input type="checkbox"/>	<input type="checkbox"/>		
adresa	VARCHAR	255	None	utf8_unicode_c		<input type="checkbox"/>	<input type="checkbox"/>		

Рисунок 3.4 – Вікно налаштування полів таблиці

**Length/Values (Довжина/Значення)** – числове поле, в якому вказується довжина текстового поля, довжина числа і т.д.

**Default** – значення за замовчуванням.

**Collation** – кодування поля.

**Attributes (Атрибути)** – для числових полів вказується беззнакове число чи ні.





**Null (Нуль)** – якщо поле не буде використовуватись можна вказати, що воно буде нульовим.

**Index (Індекс)** – обираємо параметр індексації поля (індексоване, первинне, унікальне).

**A\_I** – містить параметр AUTO\_INCREMENT. Даний параметр присвоює автонумерацію для поля. Його необхідно використовувати для ключового поля PRIMARY KEY.

**Comments** – коментар.

**MIME type** – тип даних у полі.

Назвіть перше поле “Id”, тип оберіть INT та вкажіть, що це первинний ключ (Index(PRIMARY)), включіть опцію AUTO\_INCREMENT.

Друге поле буде містити ПІБ студента, поле назвемо “Name” з типом CHAR та довжиною 100.

Третє поле буде відповідати за адресу студента, поле буде називатись “Adressa” з типом CHAR та довжиною 255.

Після завершення налаштувань, натискаємо **Save**.

Якщо все зробити вірно, то буде створено нову таблицю “Student”, яка відобразиться в лівій частині вікна.

### 3. ОСНОВНІ ОПЕРАЦІЇ З ДАНИМИ В PHPMYADMIN

У таблиці можна виконувати наступні операції: переглядати дані, додавати дані, видаляти дані, виконувати пошук даних.

Додавання даних виконується на закладці “insert” (рис. 3.5).

Введіть у відповідні поля дані та натисніть **Go**. Поле “id” можна залишити пустим, оскільки для нього задана автонумерація.

Переглянути введені дані можна на закладці “browse”(рис. 3.6), система сформує запит на перегляд даних (SELECT \* FROM `Student` LIMIT 0, 30).

Column	Type	Function	Null	Value
id	int(11)			
name	varchar(100)			Іваненко Іван Іванович
adresa	varchar(255)			м.Рівне, вул. Студентська, 6

Рисунок 3.5 – Діалог додавання запису в таблицю

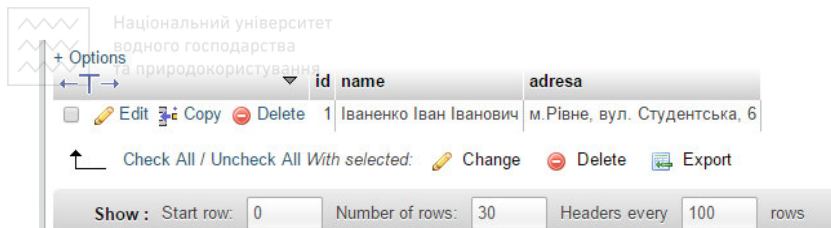


Рисунок 3.6 – Вікно перегляду даних

У цьому вікні записи можна редагувати, копіювати та видаляти.

В phpMyAdmin є також можливість фільтрації та пошуку даних, ці дії можна виконати на закладках “SQL”, “Search”.

У разі необхідності можна також імпортувати чи експортувати таблиці на відповідних закладках, що суттєво спрощує перенесення даних із однієї бази в іншу.

## ЛАБОРАТОРНА РОБОТА № 4 ЗНАЙОМСТВО З МОВОЮ PHP

1. Опис мови програмування PHP
2. Особливості мови PHP
3. Робота з рядками
4. Змішані типи даних
5. Конструкції мови
6. Базовий синтаксис

PHP (англ. PHP: Hypertext Preprocessor – PHP: гіпертекстовий препроцесор), попередньо: Personal Home Page Tools – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-серверу. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. Проект за яким був створений PHP – проект з відкритими програмними кодами.

PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта. На відміну від таких скриптових мов програмування, як JavaScript, користувач не має доступу до PHP-коду, що є перевагою з точки зору безпеки але значно погіршує інтерактивність сторінок. Але ніщо не забороняє



використовувати PHP для генерування і JavaScript-кодів які виконаються вже на стороні клієнта.

## 1. ОПИС МОВИ ПРОГРАМУВАННЯ PHP

З точки зору системи типізації, PHP є мовою програмування з динамічною типізацією. Немає необхідності явного визначення типу змінних, хоча така можливість існує. У разі звернення до змінної, інтерпретатор PHP трактує її тип відповідно до контексту. За необхідності можливе приведення змінної до певного типу за допомогою відповідних конструкцій мови. Це може знадобитись, якщо зважити, що значення змінної можуть трактуватись по-різному в залежності від її типу. Також можливе визначення типу відповідної змінної на певному етапі виконання сценарію. Імена змінних чутливі до регістру символів.

До базових типів належать булеві дані, цілі та дійсні числа із плаваючою комою, а також рядки. Булеві дані виражають істинність значення. Цілі числа можуть бути подані у вісімковому, десятковому та шістнадцятковому вигляді. Розмір цілого числа може змінюватись залежно від платформи, як правило, розрядність становить 32 біти. PHP не підтримує беззнакові цілі числа. Дійсні числа із плаваючою комою можуть бути подані в десятковій або експоненційній формі.

## 2. ОСОБЛИВОСТІ МОВИ PHP

PHP – мова, яка може бути вбудована безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP-інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустрине парну екрануючу послідовність (?>).

Велика різноманітність функцій PHP дають можливість уникнути написання багаторядкових призначених для користувача функцій на мовах програмування C або Pascal.

В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase. Через стандарт відкритого інтерфейсу зв'язку з базами даних (Open Database Connectivity Standard – ODBC) можна підключатися до всіх



баз даних, для яких існує драйвер.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP – мова, що поєднує переваги Perl і C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що на даний момент є мало не найпопулярнішою мовою для створення web-додатків (скриптів).

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили благотворний вплив на багато проєктів, в першу чергу – Linux, хоч і успіх проєкту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проєкту PHP. Ухвалення стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціниму послугу користувачам. Додатково, користувачі PHP по всьому світу є свого роду колективною службою підтримки, і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте, щоб не робили розробники PHP, виконувані файли, отримані за допомогою компіляції, працюватимуть значно швидше – в десятки, а іноді і в сотні разів. Але продуктивність PHP цілком достатня для створення цілком серйозних web-додатків.

### 3. РОБОТА З РЯДКАМИ

Рядки ділять на два класи – рядки, що підлягають аналізу, та ті, що не підлягають. Перший клас досліджується інтерпретатором на наявність посилань на інші змінні, і за умови їхньої наявності робиться підстановка значень у відповідне місце. Крім того, цей



клас дозволяє проводити маніпуляції з керуєчими символами. Символ рядка може мати лише одне з 256 значень, але є можливість працювати з багатобайтовими символами. Доступ до символів рядка можливий з використанням синтаксису, схожого на доступ до елементів масивів.

PHP надає широкий спектр функцій для пошуку та заміни тексту в рядках. Для цього використовують як традиційний підхід, так і спеціальний, що базується на використанні регулярних виразів. При цьому в мові реалізована підтримка двох видів регулярних виразів – Perl-сумісні та POSIX-сумісні, що розрізняються за синтаксисом та особливостями роботи.

## 4. ЗМІШАНІ ТИПИ ДАНИХ

До змішаних типів належать масиви, хеші та об'єкти. Масиви в сенсі мови є наборами змінних, що згруповані в єдину змінну. Вимога однотипності наповнення масивів не ставиться. Технічно, масиви – це сукупність елементів одного типу даних, впорядкованих за індексами, які зазвичай репрезентовані натуральними числами, що визначають положення елемента в масиві. Вміст значення, на яке вказує індекс (ключ) може бути будь-чим, що можна подати у вигляді змінної.

Особливістю мови є відмова від рівномірного розподілу ключів масивів. Реалізовано і модель багатовимірних масивів, причому без явного обмеження глибини вкладеності. Корисною властивістю PHP є можливість асоціації масивів із функцією зворотного виклику. Ці функції дозволяють проводити дії над одним чи кількома масивами в пакетному режимі.

Також, існують два спеціальні типи даних – ресурс та NULL. Ресурс – спеціальна змінна, що містить посилання на зовнішній ресурс. Ресурси створюються та використовуються в спеціалізованих функціях. Оскільки тип містить спеціальні вказівники на відкриті файли, під'єднання та інше, то будь-які дії над значенням ресурсу не мають сенсу.

Область видимості змінної – середовище, в якій вона визначена. Розрізняють локальні та глобальні змінні. За замовчуванням, всі змінні мають локальний характер дії. Виділяють особливий тип змінних – статичні, що дозволяє повторне звернення до змінної в певному сегменті коду, причому змінна буде зберігати попередньо



отримане значення. Існує також поняття суперглобальних змінних, які є місцем збереження даних оточення або даних, отриманих ззовні. Підтримується концепція динамічних змінних та функцій.

Константи в РНР – ідентифікатори простих значень. Можливе визначення константи, причому після її оголошення стає неможливою зміна її значення чи анулювання. Константи можуть мати лише скалярні значення. Підтримується можливість отримання значення константи за динамічним ім'ям. Область видимості констант буде глобальною для сценарію та всіх під'єднаних компонентів. Також в ядрі мови визначено чимало системних констант.

## 5. КОНСТРУКЦІЯ МОВИ

Оператори в сенсі мови дозволяють виконувати відповідну дію над одним чи кількома операндами. Оператори бувають трьох типів – унарні, бінарні та тернарні. Оператори, як і в інших мовах характеризуються не лише дією, а й асоціативністю та пріоритетністю.

Округлення відбуваються завжди в меншу сторону. В мові реалізовані особливі класи операторів – виконання, управління помилками та перевірки приналежності до класу.

Функції в сенсі мови є контейнерами коду, причому можливе включення інших функцій та класів. На цьому і базується можливість умовного визначення функції. В цьому випадку висувається вимога попередньої декларації викликаної функції, що не обов'язкове в інших випадках. Можливості перевизначення чи деактивації функції не існує. Результат, який повертає функція може мати будь-який тип.

У мові реалізована функціональність посилань. Можливо створити скільки завгодно псевдонімів, що посилаються на єдиний сегмент даних. При вивільненні будь-якого з псевдонімів, сегмент даних залишається в пам'яті до моменту завершення сценарію або вивільнення усіх посилань.

Що стосується функцій в РНР, то замість прийнятого в багатьох мовах принципу перезавантаження функцій, що дозволяє змінити хід виконання певної функції в залежності від типу та кількості переданих параметрів, використовується метод динамічних аргументів. Це дає змогу не визначати кількість параметрів для



функцій при їх оголошенні, а працювати із тими аргументами, які були отримані на момент виклику функції. У тілі функції можливо отримати кількість переданих їй аргументів і проводити відповідні маніпуляції. При оголошенні функції звичайним чином, можливе задання значень аргументів за замовчуванням. Функції можуть повертати лише одне значення, проте це обмеження можна оминати, використавши не лише масиви, а й посилання. Передача аргументів за посиланням неможлива під час виконання та оголошення функції.

Після виконання сценаріїв, простір пам'яті, займаної ними, очищується збирачем сміття. Проте, за необхідності можливе виконання очищення пам'яті від надлишкових сегментів даних під час виконання скриптів. Використання функцій очищення пам'яті є невиправданим, хоча така можливість існує.

Для побудови програмних комплексів можна використовувати модульний підхід, виконуючи розділення різнорідного коду. При потребі, можливе виконання під'єднання необхідних модулів, причому операція виконання може бути і умовною. Під'єднані до скрипта файли можуть повертати значення.

## 6. БАЗОВИЙ СИНТАКСИС природокористування

### Відокремлення PHP коду від HTML розмітки

Всі сценарії оформляються у вигляді блоків коду. Ці блоки можуть бути поміщені в HTML-код, але відділені від нього відповідними обмежувачами. Код PHP в HTML повинен знаходитись між початковим тегом `<?php` та кінцевим `?` (або між `<script language="php">` та `</script>`) Бажаним варіантом виділення PHP коду є варіант `<?php ?>`, оскільки саме такі початковий та кінцевий теги дозволять використовувати PHP код в документах, які відповідають правилам XML. Також можна користуватися скороченим записом: `<? ?>` (в `php.ini` змінна `short_open_tag` повинна мати значення `On`) і записом в стилі ASP: `<% %>` (в `php.ini` змінна `asp_tags` повинна мати значення `On`). Проте стиль ASP не рекомендується і очікується, що він буде відсутній у PHP6.

### Відокремлення інструкцій

Інструкції в PHP відокремлюються символом `;`. Перед закінченням скрипту (перед тегом `?>`) крапку з комою ставити необов'язково.



## Коментарі

Коментарі в PHP можуть бути як однорядкові так і багаторядкові. Однорядкові коментарі починаються із // або # . Багаторядкові коментарі починаються символами /\* і закінчуються \*/ . Слід зазначити, що однорядкові коментарі йдуть або до кінця поточного рядка, або до завершального тега ?>

## Змінні

Основою будь-якого програмування є змінні. Існує три стилі оформлення змінних у PHP:

- Короткий стиль:

`$variable` Використовується в більшості випадків. Змінна, яка створюється програмним кодом, записується таким чином. Можливе також використання такого стилю для отримання змінної із ФОРМИ, якщо ввімкнено `register_globals` у файлі конфігурації `php.ini`.

- Середній стиль:

`$_POST['variable']`, `$_GET['variable']`, `$_REQUEST['variable']` Використовується для отримання даних з полів ФОРМИ. В залежності від типу передачі даних, встановлюється POST або GET або REQUEST для обох випадків.

- Довгий стиль:

`$HTTP_POST_VARS['variable']`, `$HTTP_GET_VARS['variable']` Найбільш повний стиль запису, найменш використаний. У майбутньому буде вважатись застарілим, отже, небажаний.

Змінні, які обумовлюються користувачем, записуються у вигляді:

`$variable = 0.00;`, `$variable1 = 'hello PHP-Master!'`

## Застосування мови програмування PHP

Спочатку визначимо сферу застосування PHP. Як і будь-яка інша скриптова мова, PHP використовується на сайтах, зміст (контент) яких є динамічним, тобто залежить від того, хто, коли і яким чином його переглядає, а також має можливість доповнювати та редагувати контент (портали, форуми, гостьові книги). Також PHP може стати у пригоді для організації статичних сайтів великого об'єму, що містять велику кількість документів, які потрібно структурувати та привести до подібного вигляду (електронні бібліотеки, сайти документації).





## ЛАБОРАТОРНА РОБОТА № 5

# СТВОРЕННЯ ІНТЕРАКТИВНИХ ФОРМ ДЛЯ ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ

1. Створення інтерактивної форми для введення даних.
2. Реалізація виведення даних із бази.

Створювати форми для введення даних та реалізовувати виведення даних для перегляду, будемо за допомогою програми Dreamweaver.

## 1. СТВОРЕННЯ ІНТЕРАКТИВНОЇ ФОРМИ ДЛЯ ВВЕДЕННЯ ДАНИХ

Запускаємо програму Dreamweaver.

Обираємо в стартовому вікні “Создать php”(рис. 5.1).

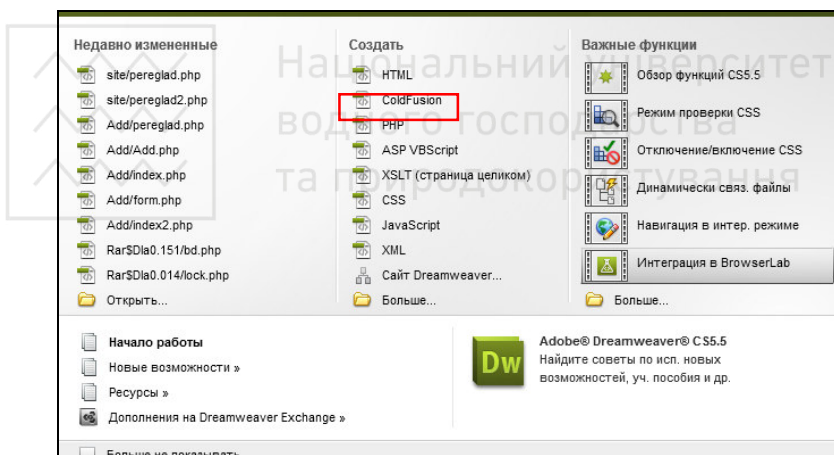
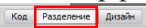


Рисунок 5.1 – Стартовое вікно

Після запуску вікна програми, в правому верхньому кутку обираємо, вид інтерфейсу “Класичний”. Обираємо вид представлення .

У блоці коду вводимо назву сторінки (рис. 5.2).

Далі, додаємо елементи форми на сторінку. А саме: поля, які були створенні у базі (ПІБ студента, адреса), кнопку "Додати студента".



У меню вставки переходимо на закладку “Форми” та обираємо потрібні елементи, потім клацаємо на сторінці розміщуючи їх.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Додавання даних про студента</title>
6 </head>
7
8 <body>
9 </body>
10 </html>

```

Рисунок 5.2 – Код початкової сторінки

Під час додавання поля з’являється наступний діалог (рис. 5.3).

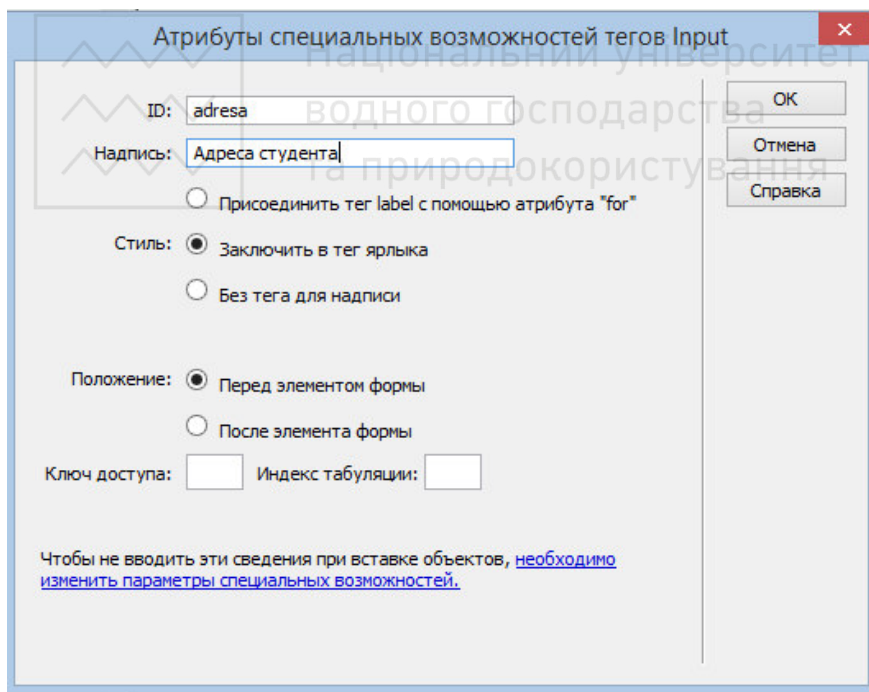


Рисунок 5.3 – Вікно додавання поля



Далі обираємо форму, клацаючи по червоній штриховій лінії та у її властивостях додаємо назву програми обробника, програми, яка буде передавати дані із форми в БД (позиція 1 на рис. 5.4). В нашому випадку вводимо “*add\_student.php*”.

Одержимо форму наступного виду (позиція 2 на рис. 5.4).

Зберігаємо документ з формою під назвою “*index.php*”.

Тепер створюємо файл обробник, який буде додавати дані у базу.

Виконуємо команду **Файл – Створити – php**. Зберігаємо файл під назвою “*add\_student.php*” в тій самій папці, що і попередній файл.

Після цього, у файлі прописуємо параметри підключення до сервера, бази даних, вибору таблиці, та SQL запит на додавання даних.

Вміст файлу *add\_student.php* (додавання даних в таблицю Students) показано на рис. 5.5.

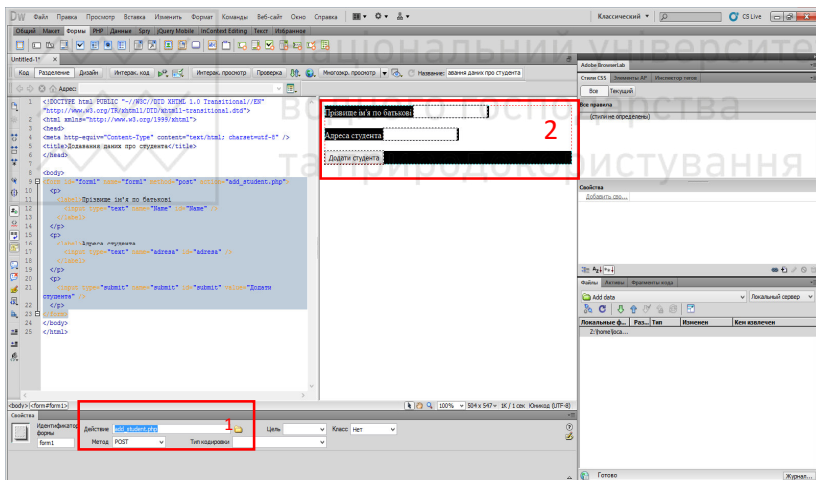


Рисунок 5.4 – Вигляд новоствореної форми.

## 2. РЕАЛІЗАЦІЯ ВИВЕДЕННЯ ДАНИХ ІЗ БАЗИ

Для перегляду введених даних, створимо ще один php-документ, назвавши його *inf\_student.php*, збережемо його там же, де й попередній.

У документі запишемо ті ж параметри для підключення до бази даних, що і в попередньому скрипті. SQL-запит на відміну від

попереднього документу буде на вибірку даних. Виведення даних реалізуємо через цикл do...while.

Приклад коду можна переглянути на рис. 5.6.

```

1  <?
2  /*Присвоєння змінним даних із форми*/
3  if (isset($_POST['name'])) {$name = $_POST['name'];
4  if ($name == '') {unset($name);} }
5  if (isset($_POST['adresa'])) {$adresa = $_POST['adresa'];
6  if ($adresa == '') {unset($adresa);} }
7
8  /*Підключення до сервера*/
9  $db = mysql_connect("mysql.hostinger.com.ua","u538371819_nauka","123456");
10 /*$db = mysql_connect("Адреса сервера MySQL","Ім'я користувача","Пароль");*/
11
12 /*Вибір БД*/
13 mysql_select_db("u538371819_nauka",$db);
14 /*mysql_select_db("Назва БД",Назва підключення)*/
15
16 /*Запит на додавання даних в БД*/
17 $result = mysql_query("INSERT INTO Student(name,adresa) VALUES ('$name','$adresa')",$db);
18
19 /*Повідомлення про результат додавання даних*/
20 if ($result = 'true') {echo "<р>Інформацію про студента додано,
21 <a href='inf_student.php'>переглянути результати</a></р>";}
22 else {echo "<р>Інформацію не додано</р>";}
23 ?>

```

Рисунок 5.5 – Скрипт додавання даних в таблицю

```

1  <?
2  /*Підключення до сервера*/
3  $db = mysql_connect("mysql.hostinger.com.ua","u538371819_nauka","123456");
4  /*$db = mysql_connect("Адреса сервера MySQL","Ім'я користувача","Пароль");*/
5
6  /*Вибір БД*/
7  mysql_select_db("u538371819_nauka",$db);
8  /*mysql_select_db("Назва БД",Назва підключення)*/
9
10 /*Запит на вибірку даних*/
11 $result = mysql_query("SELECT id,name,adresa FROM Student",$db);
12
13 /*Перенесення даних до масиву*/
14 $myrow = mysql_fetch_array ($result);
15
16 /*Виведення даних у циклі*/
17 do {
18 printf("<table width='600' border='0' align='center'>
19 <tr>
20 <td width='100'>§s</td>
21 <td width='200'>§s</td>
22 <td width='300'>§s</td>
23 </tr>
24 </table>",$myrow["id"],$myrow["name"],$myrow["adresa"]);
25 }
26 while ($myrow = mysql_fetch_array ($result));
27
28 /*Виведення посилання для додавання нового рядка даних*/
29 echo("<a href='index.php'>Додати інформацію про наступного студента</a>");
30 ?>

```

Рисунок 5.6 – Код на виведення даних із таблиці



## РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА РЕСУРСИ

1. Forum: PHP Development [Електронний ресурс]. – Режим доступу: <http://forums.devshed.com/php-development-5/>.
2. MySQL Documentation: MySQL Reference Manuals [Електронний ресурс]. – Режим доступу : URL: <http://dev.mysql.com/doc/>.
3. Php.Documentation [Електронний ресурс]. – Режим доступу: URL: <http://php.net/docs.php>.
4. Створення бази даних в phpMyAdmin [Електронний ресурс]. – Режим доступу: <http://programming.in.ua/web-design/allphp/139-createdatabase.html>.
5. Хостинг (матеріал Вікіпедії) [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/Хостинг>.





## ДОДАТОК ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

---

**Аккаунт** (англ. account) – сукупність даних, яка описує користувача. На відміну від логіна, який лише ідентифікує його, аккаунт містить більше інформації, що відноситься до користувача. У кожному конкретному випадку це може бути різний набір даних.

**Алгоритм** (лат. algorithmi) – послідовність, система, набір систематизованих правил виконання обчислювального процесу, що приводить до розв'язання певного класу задач після скінченного числа операцій. При написанні комп'ютерних програм алгоритм описує логічну послідовність операцій.

**Булевий тип даних** – див. логічний (булевий) тип даних.

**Веб-сайт** або **сайт** (англ. website – місце, майданчик в інтернеті) – сукупність веб-сторінок, доступних у мережі Інтернет, які об'єднані як за змістом, так і навігаційно. Фізично сайт може розміщуватися як на одному, так і на кількох серверах.

**Веб-сторінка** (англ. web-page) – інформаційний ресурс доступний в мережі Інтернет, який можна переглянути у веб-браузері.

**Відкрите програмне забезпечення** (англ. Open-source software) – програмне забезпечення з відкритим вихідним кодом. Вихідний код таких програм доступний для перегляду, вивчення та зміни, що дозволяє користувачеві взяти участь у доопрацюванні програми, використовувати код для створення нових програм та виправлення в них помилок.

**Дані** – інформація, подана у формалізованому вигляді, прийнятному для обробки автоматичними засобами за можливої участі людини.

**Домен** (англ. domain) – частина простору ієрархічних імен мережі Інтернет, що обслуговується групою серверів доменних імен (DNS-серверів) та централізовано адмініструється. Кожний домен має унікальне ім'я, а кожен комп'ютер, підключений до Інтернету, має, як правило, доменне ім'я. Домени мають між собою ієрархічні відношення. Говорячи простими словами, домен – це основна частина адреси сайту. *З термінології баз даних домен* – обмежена підмножина значень даного типу, яка визначає набір або діапазон допустимих значень для поля;



**Доменная система імен** (англ. Domain Name System, DNS) – ієрархічна розподілена система перетворення імені хоста (комп'ютера або іншого мережевого пристрою) в IP-адресу (і, при необхідності, навпаки). DNS-сервери зберігають інформацію про вузли, імена яких належать домену і виконують трансляцію їх імен в адреси.

**Драйвер** (англ. driver) – комп'ютерна програма, за допомогою якої операційна система отримує доступ до апаратного забезпечення обладнання.

**Застосунок** або **прикладна програма** (англ. application, application software) – користувацька комп'ютерна програма, що дає змогу вирішувати конкретні прикладні задачі користувача.

**Змінна** (англ. variable) – у *програмуванні* об'єкт програми, що має ім'я та значення. В процесі компіляції програми для змінних виділяється пам'ять конкретного розміру, яка зберігається за змінними на весь період роботи програми. Такі змінні є **статичними**, доступ до відповідних областей пам'яті здійснюється за їх іменами. Змінні, які з'являються в процесі роботи програми або їх розмір може змінюватися при її роботі називають **динамічними** змінними. Динамічна змінна явно не описується, до неї немає доступу по імені. Замість цього використовується спеціальна змінна – вказівник, значенням якого є адреса пам'яті, де розміщується динамічний об'єкт.

**Змінна динамічна** – див. змінна.

**Змінна статична** – див. змінна.

**Ідентифікатор** або **ID** (англ. data name, identifier) – унікальна ознака об'єкта, що дозволяє відрізнити його від інших об'єктів.

**Інструкція** (англ. statement) – у *програмуванні* команда мови програмування, якою задається певний крок процесу обробки інформації на ЕОМ. Програма зазвичай складається з послідовності інструкцій.

**Інтернет-провайдер** (англ. Internet service provider, ISP – постачальник інтернет-послуги) іноді просто **провайдер** – організація, що надає послуги доступу до мережі Інтернет та інші пов'язані з Інтернетом послуги.

**Інтерпретатор** (англ. interpreter) – програма, що виконує інтерпретацію.



**Інтерпретація** (англ. interpretation) – пооператорний (покомандно, порядковий) аналіз, обробка і тут же виконання вихідної програми або запиту (на відміну від компіляції, при якій програма трансляється без її виконання)

**Інтерфейс** (англ. interface) – сукупність можливостей, способів і методів взаємодії двох систем, пристроїв або програм для обміну інформацією між ними, визначена їх характеристиками, характеристиками з'єднання, сигналів обміну тощо.

**Клієнт** (англ. client) – комп'ютер (або програма), що використовує ресурси, надані іншим комп'ютером (або програмою), який називається сервером.

**Ключ** – мінімальний набір атрибутів, за значеннями яких можна однозначно знайти необхідний екземпляр об'єкту.

**Ключ зовнішній** – поле підпорядкованої таблиці, необхідне для зв'язку між таблицями.

**Ключ первинний** або **ключове поле** (англ. primary key) – поле, що однозначно ідентифікує запис.

**Коментар** – пояснення до початкового тексту програми, що знаходиться безпосередньо усередині коментованого коду. Синтаксис коментарів визначається мовою програмування. З точки зору компілятора або інтерпретатора, коментар – частина тексту програми, що не впливає на її семантику.

**Компілятор** (англ. compiler від англ. to compile – збирати в ціле) – комп'ютерна програма (або набір програм), що перетворює (компілює) сирцевий код, написаний певною мовою програмування, на семантично еквівалентний код в іншій мові програмування, який, як правило, необхідний для виконання програми машиною, наприклад, комп'ютером. Коротко компілятор можна визначити, як програму або технічний засіб, що виконує компіляцію.

**Компіляція** (лат. compilatio) – трансляція вихідного коду програми в бінарний код.

**Константа** (лат. constans – стала величина) – величина, що не змінює свого значення протягом певного процесу.

**Контент** (англ. content – зміст) будь-яке інформаційно значуще або змістовне наповнення інформаційного ресурсу або веб-сайту.

**Логічний (булевий) тип даних** – примітивний тип даних в інформатиці, що може приймати два можливих значення, що іноді називаються істиною і хибістю (також так й ні).





**Масив** (англ. array) – впорядкований набір фіксованої кількості однотипних елементів, що зберігаються в послідовно розташованих комірках оперативної пам'яті, мають порядковий номер і спільне ім'я, що надає користувач. У програмуванні сукупність елементів одного типу даних, впорядкованих за індексами, які зазвичай репрезентовані натуральними числами, що визначають положення елемента в масиві.

**Мережа комп'ютерна** – об'єднання комп'ютерів та інших пристроїв лініями зв'язку таким чином, що комп'ютери (або їх користувачі) можуть обмінюватись ресурсами один з одним.

**Мова програмування** (англ. programming language) – штучна мова, створена для передачі команд машинам, зокрема комп'ютерам. Мови програмування використовуються для створення програм, котрі контролюють поведінку машин та запису алгоритмів.

**Область видимості змінної** – контекст, у рамках якого змінна була визначена і де до неї можна отримати доступ. Позначає область програми, в межах якої ідентифікатор (ім'я) деякої змінної продовжує бути пов'язаним з цією змінною і повертати її значення. За межами області видимості той же самий ідентифікатор може бути пов'язаний з іншою змінною, або бути вільним (не пов'язаним ні з якою з них).

**Операнд** (англ. operand) – аргумент операції; дані, які обробляються командою; граматична конструкція, яка позначає вираз, що задає значення аргументу операції; іноді операндом називають місце або позицію в тексті, де має стояти аргумент операції.

**Оператор** (англ. operator) – у програмуванні спеціальний символ, який повідомляє транслятору про те, що ви хочете виконати операцію з деякими операндами (наприклад, +, -, %, <<).

**Піддомен** (англ. subdomain) або **субдомен** – домен, що є частиною домену вищого рівня; дочірній домен по відношенню до основного.

**Підпрограма** (англ. subroutine) – частина програми, яка реалізує певний алгоритм і дозволяє звернення до неї з різних частин загальної (головної) програми. В термінах мов програмування: функції (C), процедури (Pascal), методи (C++, Java, C#).

**Провайдер** – див. інтернет-провайдер.



**Програма комп'ютерна** (англ. computer program) – набір інструкцій у вигляді слів, цифр, кодів, схем, символів чи у будь-якому іншому вигляді, виражених у формі, придатній для зчитування (комп'ютером), які приводять його у дію для досягнення певної мети або результату; низка команд для комп'ютера, що становлять запис алгоритму однією з мов програмування.

**Протокол передачі даних** – набір угод інтерфейсу логічного рівня, які визначають обмін даними між різними програмами. Ці угоди задають однаковий спосіб передачі повідомлень і обробки помилок при взаємодії програмного забезпечення рознесеного в просторі обладнання, з'єданого тим чи іншим інтерфейсом.

**Процедура** – у програмуванні див. підпрограма

**Сайт** – див. веб-сайт.

**Сервер** (англ. server – обслуговуючий) – 1. як комп'ютер – комп'ютер у локальній чи глобальній мережі, який надає користувачам свої обчислювальні і дискові ресурси, а так само доступ до встановлених сервісів; найчастіше працює цілодобово, чи у час роботи групи його користувачів; 2. як програма – програма, що надає деякі послуги іншим програмам (клієнтам).

**Синтаксис мови програмування** – набір правил, який визначає те, як буде виглядати програма на цій мові, зокрема, як пишуться оператори, оголошення і інші мовні конструкції.

**Скрипт** або **сценарій** (англ. script) – програма, яка автоматизує деяке завдання, яке без сценарію користувач робив би вручну, використовуючи інтерфейс програми; короткі описи дій, виконуваних системою.

**Скриптова мова** (англ. scripting language) – мова програмування, розроблена для запису «сценаріїв», послідовностей операцій, які користувач може виконувати на комп'ютері.

**Субдомен** – див. піддомен.

**Сценарій** – див. скрипт.

**Тип даних** – характеристика, яка визначає множину припустимих значень, формат їхнього збереження, розмір виділеної пам'яті та набір операцій, які можна робити над даними.

**Функція** – у програмуванні див. підпрограма

**Хост** (англ. host – господар, який приймає гостей) або **вузол** – будь-який пристрій, що містить ресурс і надає до нього доступ. У вузькому значенні під хостом розуміють будь-який комп'ютер,



під'єднаний до мережі. Також під поняттям **хост** можуть мати на увазі унікального відвідувача сайту.

**Хостинг** – послуга, що надає дисковий простір для розміщення фізичної інформації на сервері, що постійно перебуває в мережі (наприклад Internet).

**ASP** (англ. Active Server Pages – активні серверні сторінки) – технологія від компанії Microsoft, що дозволяє динамічно формувати автоматично оновлювані веб-сторінки з боку веб-сервера.

**CGI** (англ. Common Gateway Interface – загальний інтерфейс шлюзу) – стандарт інтерфейсу, який використовується для зв'язку зовнішньої програми з веб-сервером. Програму, яка працює за таким інтерфейсом спільно з веб-сервером, прийнято називати шлюзом, хоча використовуються назви «скрипт» (сценарій) або «CGI-програма». Сам інтерфейс розроблений таким чином, щоб можна було використовувати будь-яку мову програмування, яка може працювати зі стандартними пристроями введення-виведення.

**Cold Fusion** – сервер застосунків і інтерпретована скриптова мова програмування, створені для генерації HTML на веб-сервері і роботи з базами даних. Призначені для веб-розробників, дає можливість створювати динамічні сайти і інтернет-застосунки.

**FTP** (англ. File Transfer Protocol – протокол передачі файлів) – стандартний протокол, призначений для передачі файлів по TCP-мережам (наприклад, Інтернет). FTP часто використовується для завантаження веб-сторінок та інших документів з приватного пристрою на відкриті сервера хостингу. При розгляді FTP як сервісу Інтернет мають на увазі не просто протокол, а саме *сервіс* – доступ до файлів, які знаходяться у файлових архівах.

**HTML** (англ. HyperText Markup Language – мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті. Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

**ID** – див. ідентифікатор.

**IP-протокол** (англ. Internet Protocol – інтернет протокол, міжмережвий протокол) – протокол мережевого рівня для передачі даних між мережами.

**Java** – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент

платформи Java. Синтаксис мови багато в чому схожий на C та C++. Програми Java зазвичай компілюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній машині незалежно від комп'ютерної архітектури.

**JavaScript** – прототипно-орієнтована сценарна мова програмування.

**ODBC** (англ. Open DataBase Connectivity – відкритий зв'язок з базами даних) – відкритий інтерфейс доступу до баз даних, незалежний від систем баз даних і операційних систем.

**Open-source software** – див. відкрите програмне забезпечення.

**SQL** (англ. Structured Query Language – мова структурованих запитів) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції.

**TCP** (англ. Transmission Control Protocol – протокол керування передачею) – один з основних мережевих протоколів Інтернету, призначений для управління передачею даних в мережах і підмережах TCP/IP.

**TCP/IP** (англ. Transmission Control Protocol / Internet Protocol – протокол керування передачею / міжмережевий протокол) – набір мережевих протоколів передачі даних, що використовуються в мережах, включаючи мережу Інтернет. Назва TCP/IP утворюється з двох найважливіших протоколів сімейства – Transmission Control Protocol (TCP) і Internet Protocol (IP), які були розроблені та описані першими в даному стандарті.

**XHTML** (англ. Extensible Hypertext Markup Language – Розширювана мова розмітки гіпертексту) – мова розмітки, що повторює і розширює можливості HTML, але відповідає синтаксичним правилам XML.

**XML** (англ. eXtensible Markup Language – розширювана мова розмітки) – запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.