МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Прктична робота № 1**

з дисципліни « Web-програмування мовою PHP »

*назва дисципліни*

на тему: «ЗНАЙОМСТВО З PHP»

Виконав: студент 3 курсу групи № 632п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Зайченко Ярослав Ігорович

(прізвище й ініціали студента)

Прийняв: старший викладач

Дем'яненко Владислав Анатолійович

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків – 2024

**ЗМІСТ**

[Постановка завдання 3](#_Toc179963230)

[Порядок виконання роботи 3](#_Toc179963231)

[Зміст звіту 3](#_Toc179963232)

[Теоретичні відомомості 4](#_Toc179963233)

[Виконання роботи 9](#_Toc179963234)

[Заключення 13](#_Toc179963235)

[*Додаток А* 14](#_Toc179963236)

[*Додаток Б* 15](#_Toc179963237)

[*Додаток В* 16](#_Toc179963238)

Мета роботи: отримання практичних навичок роботи з установки й роботі з одним з основних засобів розробки додатків на мові програмування PHP.

Постановка завдання

Навчитися встановлювати і працювати з середовищем розробки додатків на мові сценаріїв PHP (Середовище на вибір студента – Eclipse PHP).

Порядок виконання роботи

1. Прочитати теоретичний матеріал
2. Вивчити процес інсталювання EclipsePHP/PHPStrom або іншого середовища
3. Навчитися створювати проект в обраному середовищі
4. Навчитися запускати проект PHP
5. Навчитися користуватися коментарями.
6. Навчитися створювати прості програми мовою PHP
7. Зробити висновки по роботі.

Зміст звіту

1. Постановка завдання.
2. Встановлення завдань, які вирішуються в лабораторній̆ роботі
3. Описати особливості процесу створення і запуску додатка.
4. Коротко описати лексеми і висновок інформації на екран.
5. Висновки по роботі.

Теоретичні відомомості

Для створення програм на мові програмування PHP нам буде потрібно текстовий редактор. Найбільш популярним на сьогоднішній день є програма Notepad ++.

Перейдемо до раніше створеного каталогу ~\localhost, який буде зберігати всі документи сайту. Створимо текстовий файл і назвемо його index.html. Відкриємо його в текстовому редакторі і додамо в нього наступний код:

<!DOCTYPE html>  
<html>  
<head>  
<title>Перший сайт на PHP</title> <meta charset="utf-8">

</head>  
<body>  
<h2>Введіть ваші дані:</h2>  
<form action="display.php" method="POST">  
<p>Введіть ім`я: <input type="text" name="firstname" /></p> <p>Введіть прізвище: <input type="text" name="lastname" /></p> <input type="submit" value="Відправити ">  
</form>  
</body>  
</html>

Код html містить форму з двома текстовими полями. При натисканні на кнопку дані цієї форми відсилаються скрипту display.php, так як він вказаний в атрибуті action.

Тепер створимо цей скрипт, який буде обробляти дані. Додамо в папку ~\localhost новий текстовий файл. Перейменуємо його у display.php. За замовчуванням файли програм на php мають розширення .php. Отже, додамо в файл display.php наступний код:

<!DOCTYPE html>

<html>

<head>

<title>Перший сайт на PHP </title>

<meta charset="utf-8">

</head>

<body>

<?php

name = $\_POST["firstname"];  
$surname = $\_POST["lastname"];  
echo "Ваше ім`я: <b>".$name . " " . $surname . "</b>";

?>

</body>

</html>

Тут вже в розмітці html йдуть вкраплення коду PHP. Для додавання виразів PHP на сторінку використовуються теги <? Php .....?>, Між якими йдуть інструкції на мові PHP. У коді php ми отримуємо дані форми і виводимо їх на сторінку.

Кожне окреме вираз PHP має завершуватися крапкою з комою. В даному випадку у нас три вирази. Два з них отримують передані дані форми, наприклад, $ name = $ \_POST [ "firstname"] ;.

$ Name – це змінна, яка буде зберігати деяке значення. Всі змінні в PHP передує знаком $. І так як форма на сторінці index.html використовує для відправки метод POST, то за допомогою виразу $ \_POST [ "firstname"] ми можемо отримати значення, яке було введено в текстове поле з атрибутом name = "firstname". І це значення потрапляє в змінну $ name.

За допомогою оператора echo можна вивести на сторінку будь-яке значення або текст, які йдуть після оператора. В даному випадку (echo "Ваше ім'я: <b>". $ Name. "". $ Surname. "</ B>") за допомогою знака точки текст в лапках з'єднується зі значеннями змінних $ name і $ surname і виводиться на сторінку .

Основи синтаксису

При створенні першої програми на PHP вже були порушені деякі основні принципи створення скриптів на мові PHP. Тепер, розглянемо їх більш детально.

Програма або скрипт на PHP, як правило, знаходиться в файлі розширенням .php. Хоча розробники можуть також вставляти код php і в файли з розширеннями .html / .htm.

Коли користувач звертається до скрипту в адресному рядку браузера, набираючи, наприклад, http: // localhost: 8080 / display.php, то веб-сервер передає його інтерпретатора PHP. Потім інтерпретатор обробляє код і генерує на його основі html-розмітку. І потім згенерований html-код відправляється користувачеві.

Документ PHP може містити як розмітку html, так і код на мові php. Для переходу від розмітки html до коду php використовуються теги <?php та ?>, Між якими йде код php. Дані теги служать зазначенням інтерпретатора, що їх вміст треба інтерпретувати як код php, а не розмітку html.

Також можна використовувати коротку версію тегів: <? і ?>. Для цього в файлі php.ini треба змінити значення short\_open\_tag = Off на short\_open\_tag = On.

Розглянемо найпростіший скрипт на php:

<html>

<head>

<title>Веб-сайт</title>

</head>  
<body>  
<?php  
echo "<p>Привет мир!</p>"; echo "2 + 2 = " . (2+2); ?>

</body>

</html>

Після обробки файлу інтерпретатор сформує наступну розмітку:

<html>  
<head> <title>Веб-сайт</title> </head>  
<body>  
<p>Привет мир!</p> 2+2= 4  
</body>  
</html>

Тут використані дві інструкції echo "<p> Привіт світ! </ p>" і echo "2 + 2 =". (2 + 2), який виводять певне значення на сторінку. Кожна окрема інструкція в PHP завершується крапкою з комою.

Коментарі

При створенні веб-сайту ми можемо використовувати коментарі. Наприклад, ми можемо прокоментувати будь-яку дію, щоб згодом мати уявлення, що цей код робить:

<?php

echo "<p>Привет мир!</p>"; // вывод сообщения

// echo "2 + 2 = " . (2+2);  
?>

Знак // передує однорядковий коментар, і все що йде після цього знака на одному рядку, буде вважатися коментарем і не буде виконуватися інтерпретатором. При обробці інтерпретатор просто буде пропускати коментарі.

Якщо нам треба закомментувати кілька рядків, то ми можемо використовувати багаторядковий коментар / \* текст коментаря \* /:

<? php  
echo "<p> Привіт світ! </ p>"; // виведення повідомлення  
/\*  
багатостроковий коментарвисновок результату арифметичного виразу  
echo "2 + 2 =". (2 + 2);  
\*/  
?>

Всі рядки всередині коментаря також не обробляються інтерпретатором.

Змінні

Як і в багатьох мовах програмування, в PHP є змінні. Змінні зберігають окремі значення, які можна використовувати у виразах на PHP. Для позначення змінних використовується знак долара $. Наприклад:

<?php  
$a = 10;  
echo $a;  
?>

Тут визначена змінна, яка буде зберігати число 10. Присвоєння значення відбувається за допомогою знака рівності =.

Можна привласнювати значення іншої змінної:

$a = 10;

$b=$a;

echo $b;

PHP є чутливою до регістру мовою, а значить, змінні $ counter і $ Counter представлятимуть дві різні змінні.

Також при найменуванні змінних нам треба враховувати наступні правила:

* Імена змінних повинні починатися з алфавітного символу або з підкреслення
* Імена змінних можуть містити тільки символи: a-z, A-Z, 0-9, і знак підкреслення
* Імена змінних не повинні включати в себе прогалини

Перевірка існування змінної. оператор isset

Якщо змінна оголошена, але їй спочатку не присвоєно ніякого значення

(інакше кажучи вона не ініциалізованна), то нам буде проблематично її використовувати. наприклад:

<?php  
$a;

echo $a;

?>

При спробі вивести значення змінної ми отримаємо діагностичне повідомлення про те, що змінна не визначена: Notice: Undefined variable: a in C: \ localhost \ echo.php on line 3.

Оператор isset () дозволяє визначити, ініциалізованна змінна чи ні. Якщо змінна визначена, то isset () повертає значення true. Якщо змінна не визначена, то isset () повертає false. Наприклад:

<?php  
$a;  
if(isset($a))

echo $a; else

echo "переменная a не определена"; ?>

Для перевірки змінної використовувалася конструкція if ... else, яка визначає істинність виразу. І якщо вираз істинний, тоді виконується вираз після блоку if. Якщо ж вираз помилково (тобто одно false), виконується вираз після блоку else.

Тут змінна не инициализирована, тому оператор isset ($ a) буде повертати значення false, і, отже, буде спрацьовувати блок else. Якби ми присвоїли змінної деяке початкове значення, наприклад, $ a = 20, то оператор isset повернув би значення true, і браузер вивів би її значення.

За допомогою оператора unset () ми можемо знищити змінну:

<?php

$a=20;

echo $a; // 20

unset($a);

echo $a; // ошибка, переменная не определена ?>

Інтерполяція змінних в PHP

Значення змінної може бути відображено наприклад так:

<?php  
$capital = "Paris";

echo "The capital of France is**", $capital,"<**br />";

?>

Але є спосіб зробити це простіше. Якщо ім'я змінної укладено в подвійні (НЕ одинарні) лапки, то змінна інтерполюється. наприклад:

<?php

$capital = "Paris";

echo "The capital of France is **$capital** <br />";  
?>

Також існує одна тонкість при використанні інтерполяції змінних. наприклад:

<?php  
$ Text = "news";  
echo "Where's the $ textpaper <br />";  
?>

Очікується, що ім'я змінної $ text буде замінено на її значення. Але в результаті буде виведено повідомлення про помилку. Щоб цього не траплялося потрібно вказати яку частину слова textpaper ми маємо введу за ім'я змінної. Для цього потрібно взяти ту частину яка нам потрібна в фігурні дужки:

<?php  
$text = "news";  
echo "Where's the {$text}paper <br />";  
?>

Змінні, що містять імена інших змінних

У PHP можливо розміщувати в значення змінних не тільки звичайні значення, а й імена інших змінних.

<?php

$apples = 5; // Создаем переменную $apples

$fruit = "apples"; /\* Создаем переменную $fruit, которая содержит имя переменной $apples \*/

// Сейчас мы можем вывести $apples, как $$fruit  
echo "Число яблок - ", $$fruit;  
?>

Для коректного відображення подібних змінних в строкових константах, взятих в подвійні лапки, слід також використовувати фігурні дужки: ${$fruit}. наприклад:

<?php  
echo "Число яблук - ${$ fruit}";

?>

Типи даних

PHP є мовою з динамічною типізацією. Це означає, що тип даних змінної виводиться під час виконання, і на відміну від ряду інших мов програмування в PHP не треба вказувати перед змінної тип даних.

PHP підтримує вісім простих типу даних:

* boolean (логічний тип)
* integer (цілі числа)
* double (дробові числа)
* string (рядки)
* array (масиви)
* object (об'єкти)
* resource (ресурси)
* NULL

Integer (цілочисельний тип)

Являє ціле число зі знаком розміром в 32 біта (від -2 147 483 648 до 2 147 483 647).

$int = -100;

echo $int;

Тут змінна $ int представляє цілочисельний тип, так як їй присвоюється цілочисельне значення.

Крім десяткових цілих чисел PHP має можливість використовувати також виконавчі, восьмеричні і шістнадцяткові числа. Шаблони чисел для інших систем:

* шістнадцятиричні: 0 [xX] [0-9a-fA-F]
* восьмеричні: 0 [0-7]
* виконавчі: 0b [01]

Наприклад:

<?php  
// Все числа в десятичной системе имеют значение 28 $int\_10 = 28; // десятичное число  
$int\_2 = 0b11100; // двоичное число  
$int\_8 = 034; // восьмеричное число  
$int\_16 = 0x1C; // шестнадцатеричное число

echo

echo

echo

echo

?>

"int\_10 = $int\_10 <br>";

"int\_2 = $int\_2 <br>";

"int\_8 = $int\_8 <br>";

"int\_16 = $int\_16";

?>

Тип double (числа з плаваючою точкою)

Розмір числа з плаваючою точкою залежить від платформи. Максимально можливе значення, як правило, становить ~ 1.8e308 з точністю близько 14 десяткових цифр. Наприклад:

<?php

$a1 = 1.5;  
$a2 = 1.3e4; // 1.3 \* 10^4  
$a3 = 6E-8; // 0.00000006  
echo $a1 . " | " . $a2 . " | " . $a3; ?>

Тип boolean (логічний тип)

Змінні логічного типу можуть приймати два значення: true і false або інакше кажучи істина і брехня. Найчастіше логічні значення використовуються в умовних конструкціях:

<?php  
$foo = true;  
$a=10;  
$b=5;  
echo "foo = true <br>";  
if($foo)

echo $a+$b; else

echo $a-$b;

$foo = false;  
echo "<br> foo = false <br>";

if($foo)

echo $a+$b;

else

echo $a-$b;

?>

Вираз if () перевіряє істинність виразу. В даному випадку перевіряється значення змінної $ foo. Або воно істинно або дорівнює true, то виконується наступне за оператором if вираз. А якщо змінна або вираз в операторі if одно false, то виконується вираз після оператора else.

Виконання роботи

Завдання 1. Встановлення та налаштування програмного середовища для веб-розробки

WAMP (Windows Apache MySQL PHP) — комплекс програмного забезпечення для Windows систем, який дає змогу швидко розгорнути сервер для розробки або надання веб-сервісів.

Для користувачів Linux існує аналогічне програмне забезпечення – LAMP. Тепер перейдемо власне до інсталяції.

Крок1. Переходимо на сайт [www.wampserver.com/en/](http://www.wampserver.com/en/)



Рисунок 1 – Веб – сторінка WAMP

Натискаємо в меню **Download**, побачимо кілька версій програми для завантаження. Вибираємо необхідну нам збірку (в залежності від Windows та версій програм, які включені в збірку)

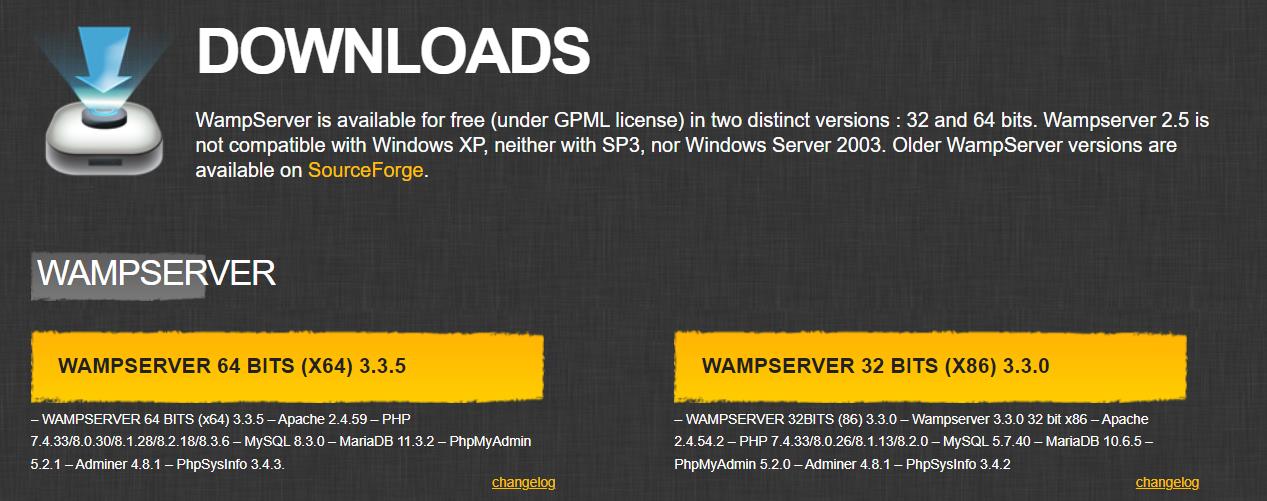


Рисунок 2 – Вибір версії ПЗ

Встановлюємо WAMP, як і всі інші програми на Windows. Можуть попросити встановити відповідні пакети для роботи.

В кінці інсталяції програма запитала браузер по-замовчуванню, для того, щоб синхронізуватись з ним та редакторв в якому планується працювати. Обираємо браузер і рухаємось далі. Після встановлення і запуску програми, перейшли за посиланням http://localhost/ , де маємо побачити відповідну сторінку, якщо все працює:

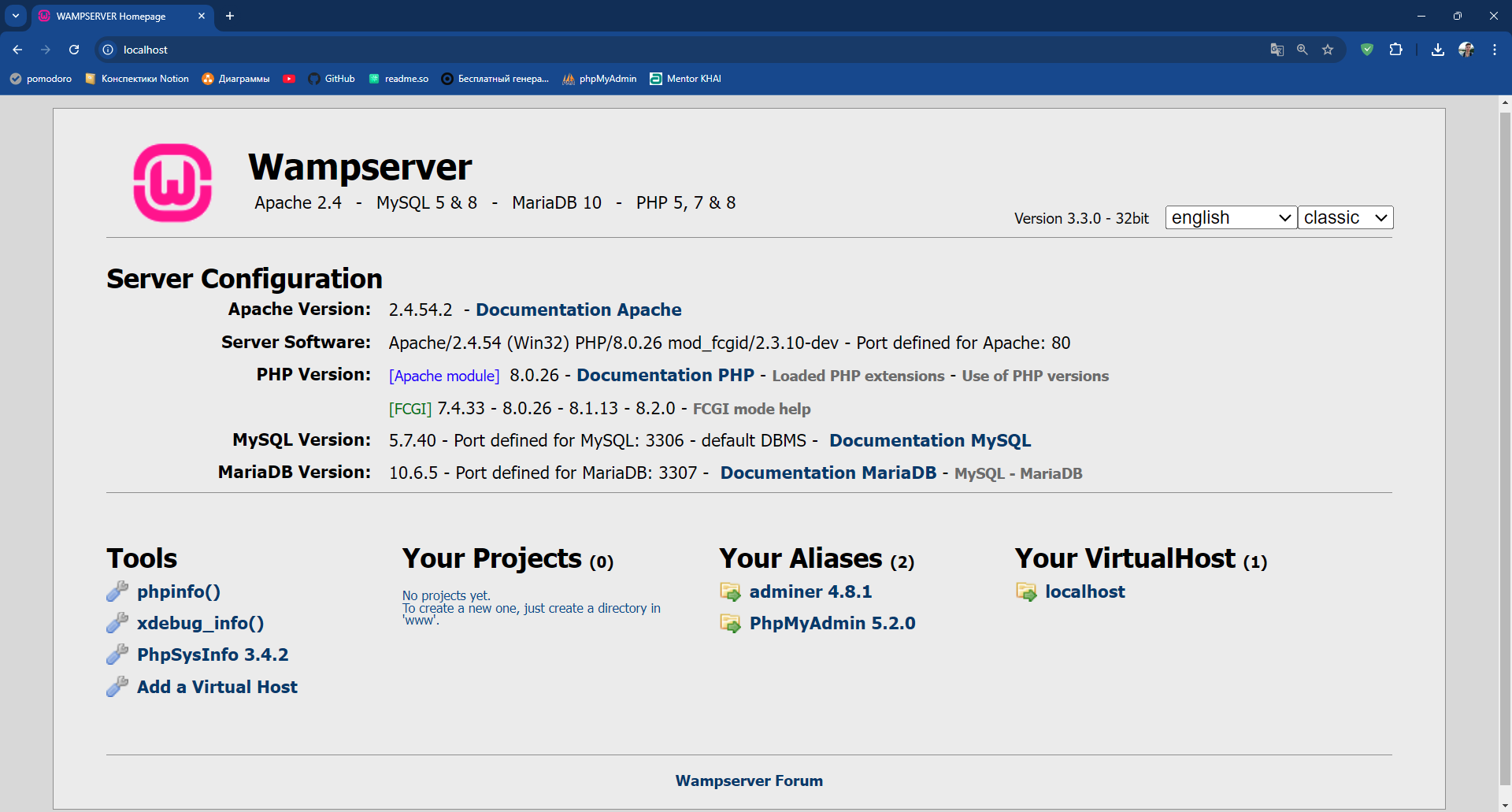


Рисунок 3 – Результат встановлення WAMP

На стартовій сторінці серверу вказані підключені модулі, є лінки на phpinfo() та phpMyAdmin, а внизу подані Your Projects, це всі проекти, які наявні на сервері. До слова, всі проекти (а це ніщо інше, як папки з файлами) повинні бути розміщені в папці wamp\www

Завдання 2. Створити текстовий документ з ім’ям index.php.

Створити першу програму, яка виводить такий текст:

*Полину в мріях в купель океану,*

*Відчую* ***шовковистість*** *глибини,*

*Чарівні мушлі з дна собі дістану,*

*Щоб* ***взимку*** *тішили мене вони...*

Код файлу поданий в Додатку А, результат роботи файлу поданий на рисунку нижче.

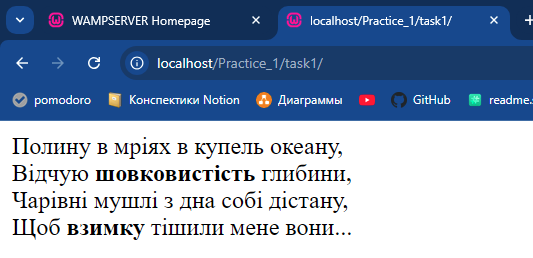


Рисунок 4 – файл index.php

Завдання для варіанту 3:

**Завданя 3.1.**

(Символ задається програмно, не вводиться з клавіатури) Написати програму, яка приймає символ (букву) і визначає, чи є він голосним чи приголосним. Використати конструкцію **switch**.

Код файлу поданий в Додатку Б, результат роботи файлу поданий на рисунку нижче.

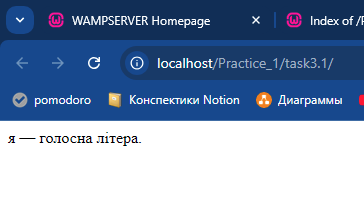


Рисунок 5 – файл index.php

**Завданя 3.2.**

Написати функцію, яка приймає число n - кількість квадратів. Функція має вивести на чорному тлі n червоних квадратів випадкового розміру в випадковій позиції вікна браузера.

Код файлу поданий в Додатку В, результат роботи файлу поданий на рисунку нижче.

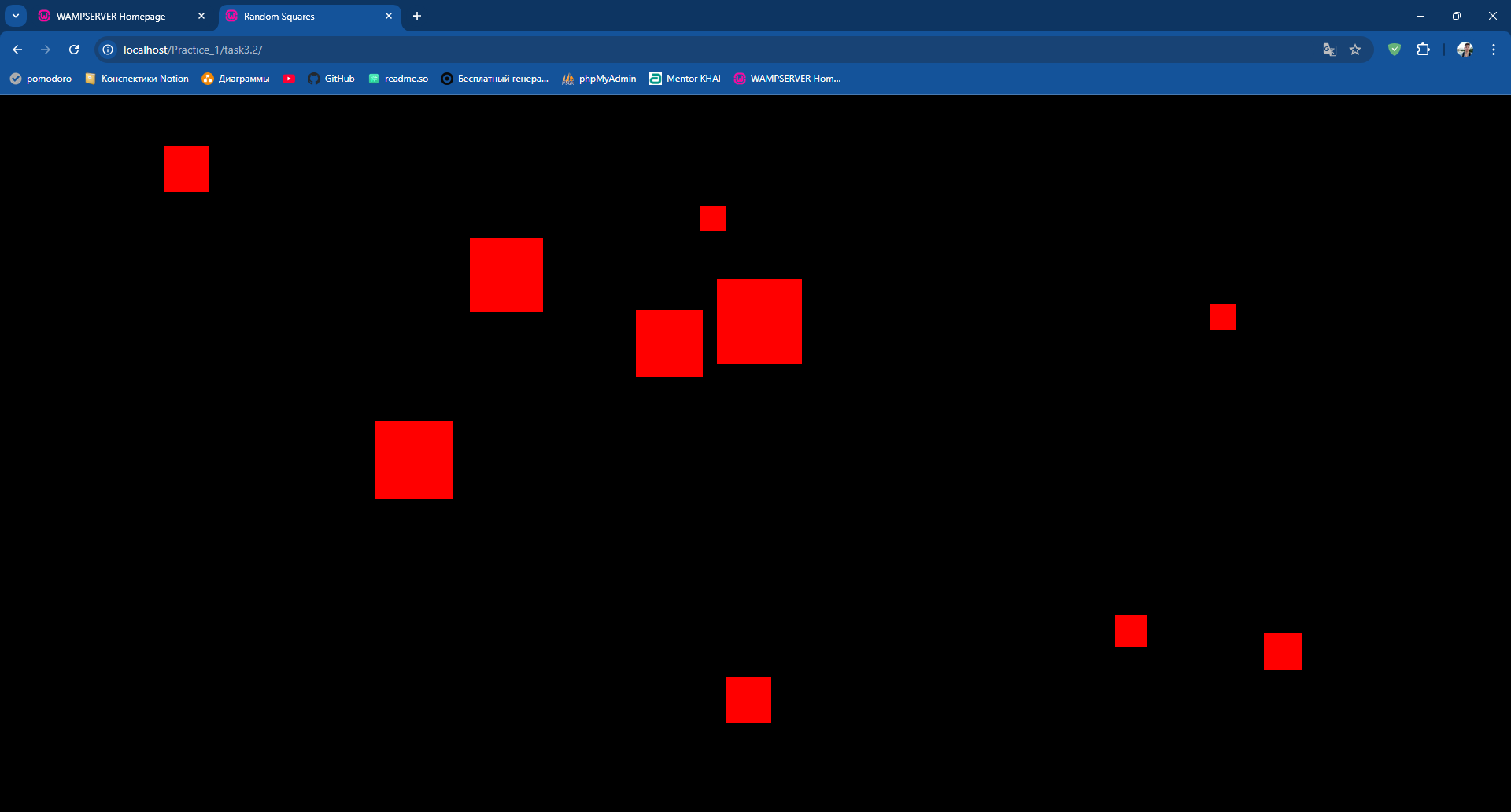


Рисунок 6 – файл index.php

Заключення

Отримано практичні навички роботи з установки і роботі з одним з основних засобів розробки додатків на мові програмування PHP в середовищі розробки Eclipse.

*Додаток А*Машинний лістинг файлу **index.php до завдання 1**

<?php

echo "Полину в мріях в купель океану,<br>";

echo "Відчую <strong>шовковистість</strong> глибини,<br>";

echo "Чарівні мушлі з дна собі дістану,<br>";

echo "Щоб <b>взимку</b> тішили мене вони...";

?>

*Додаток Б*Машинний лістинг файлу **index.php** **до завдання 3.1**

<?php

$letter = 'Я';

$letter = mb\_strtolower($letter);

switch ($letter) {

case 'а':

case 'е':

case 'є':

case 'и':

case 'і':

case 'ї':

case 'о':

case 'у':

case 'ю':

case 'я':

echo "$letter — голосна літера.<br>";

break;

// Робота з приголосними буквами

default:

echo "$letter — приголосна літера.<br>";

break;

}

?>

*Додаток В*Машинний лістинг файлу **index.php до завдання 3.2**

<?php

function generateSquares($n) {

echo "<!DOCTYPE html>";

echo "<html lang='en'>";

echo "<head>";

echo "<meta charset='UTF-8'>";

echo "<meta name='viewport' content='width=device-width, initial-scale=1.0'>";

echo "<style>";

echo "body { background-color: black; margin: 0; padding: 0; overflow: hidden; }";

echo ".square { position: absolute; background-color: red; }";

echo "</style>";

echo "<title>Random Squares</title>";

echo "</head>";

echo "<body>";

echo "<script>";

echo "const n = $n;"; // кількість квадратів

echo "for (let i = 0; i < n; i++) {";

echo " const square = document.createElement('div');";

echo " const size = Math.floor(Math.random() \* 100) + 20;"; // випадковий розмір від 20 до 120 пікселів

echo " const x = Math.floor(Math.random() \* window.innerWidth);"; // випадкова позиція по осі X

echo " const y = Math.floor(Math.random() \* window.innerHeight);"; // випадкова позиція по осі Y

echo " square.style.width = size + 'px';";

echo " square.style.height = size + 'px';";

echo " square.style.left = x + 'px';";

echo " square.style.top = y + 'px';";

echo " square.classList.add('square');";

echo " document.body.appendChild(square);";

echo "}";

echo "</script>";

echo "</body>";

echo "</html>";

}

generateSquares(10);

?>