МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ



**Звіт**

До розрахунково-графічної роботи

З дисципліни:

«Веб-технології та веб-дизайн»

Варіант №7

*Виконав:*

*студент групи КН-47*

Іванов Вадим

*Прийняв:*

*доцент* Василюк А. С.

Львів 2018

**ЗМІСТ**

1. Завдання ………………………………………………………………………….3
2. Вступ……………………………………………………………………………...4
3. Розділ 1. Теоретична частина
4. Розділ 2. Практична частина
5. Висновки
6. Список використаної літератури

**ЗАВДАННЯ**

В ході виконання розрахунково-графічної роботи потрібно в першому розділі викласти теоретичні положення про мову PHP. В другому розділі за допомогою мови гіпертекстової розмітки HTML написати ієрархічний веб-сайт з трьох сторінок на довільну тему. Також в цьому розділі потрібно детально описати процес створення цього сайту, включно з скріншотами. Код сторінки потрібно додати до роботи .

**ВСТУП**

*Актуальність теми*. З огляду на те, що зараз активно розвивається напрямок веб-розробки, з кожним роком все більше нових веб-орієнтованих проектів відкривається. Тому кожному студенту за напрямом “Комп'ютерні науки та інформаційні технології” важливим, популярним та корисним є ознайомлення з сучасними технологіями та практичне застосування певних тохнологій для написання власного веб-орієнтованого додатку.

*Мета і задачі роботи.* Метою роботи є надання коротких, проте повних теоретичних відомостей про мову PHP написання власної веб-орієнтованої аплікації. Для цього потрібно виконати наступні задачі:

* Проаналізувати літературні джерела та віднайти корисну та актуальну інформацію про мову PHP;
* Систематизувати теоретичні знання про мову PHP та викласти основні факти про цю мову та її особливості;
* Вибрати набір технологій для написання аплікації.
* Вибрати тематику для написання аплікації;
* Провести етап написання коду;
* Провести етап тестування аплікації.

**РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА**

**PHP** (Hypertext Preprocessor - PHP: [гіпертекстовий](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) [препроцесор](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80)), попередня назва: Personal Home Page Tools-[скриптова](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [мова програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), була створена для генерації [HTML](https://uk.wikipedia.org/wiki/HTML)-сторінок на стороні [веб-сервера](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із [Java](https://uk.wikipedia.org/wiki/Java" \o "Java), [.NET](https://uk.wikipedia.org/wiki/.NET), [Perl](https://uk.wikipedia.org/wiki/Perl" \o "Perl), [Python](https://uk.wikipedia.org/wiki/Python" \o "Python), [Ruby](https://uk.wikipedia.org/wiki/Ruby" \o "Ruby)). PHP підтримується переважною більшістю [хостинг](https://uk.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3" \o "Хостинг)-[провайдерів](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B2%D0%B0%D0%B9%D0%B4%D0%B5%D1%80). PHP - проект [відкритого програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F).

* 1. **Історія розвитку мови PHP**

Історія PHP починається з [1995](https://uk.wikipedia.org/wiki/1995) року, коли Расмус Лердорф ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Rasmus Lerdorf) створив простий [застосунок](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) мовою Perl, що аналізував відвідування користувачами його резюме на веб-сайті. Потім, коли цим [застосунком](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) вже користувалися кілька чоловік, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Інструменти для особистої домашньої сторінки [англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Personal Home Page Tools версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.

PHP отримав змішані відгуки через відсутність власної підтримки Unicode на рівні основної мови. У 2005 році був започаткований проект, очолюваний Андрієм Змієвським, для залучення рідної підтримки Unicode на PHP, шляхом вбудовування бібліотеки "Міжнародні компоненти для Unicode" (ICU) та вбудованих текстових рядків як UTF-16. Оскільки це призведе до серйозних змін як до внутрішньої частини мови, так і до коду користувача, планувалося випустити його як версію 6.0 мови разом з іншими основними функціями, які розвиваються.

У PHP 5 також включені два нові [модулі](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C) для роботи з [протоколами](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) - [SimpleXML](https://uk.wikipedia.org/wiki/SimpleXML) і [SOAP](https://uk.wikipedia.org/wiki/SOAP). SimpleXML дозволяє значно спростити роботу з XML-даними, представляючи вміст XML-документа у вигляді PHP-об'єкта. Розширення SOAP дозволяє будувати на PHP [сценарії](https://uk.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D1%96%D0%B9" \o "Сценарій), що обмінюються інформацією з іншими [застосунками](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA" \o "Застосунок) за допомогою [XML](https://uk.wikipedia.org/wiki/XML)-повідомлень поверх існуючих веб-протоколів, наприклад [HTTP](https://uk.wikipedia.org/wiki/HTTP). Модуль для роботи з [SOAP](https://uk.wikipedia.org/wiki/SOAP) для PHP 5 надає розробникам засіб для достатньо швидкого створення ефективних SOAP-клієнтів і SOAP-серверів.

Новий модуль PHP 5 [MySQLi](https://uk.wikipedia.org/wiki/MySQLi" \o "MySQLi) (MySQL Improved) призначений для роботи з [MySQL](https://uk.wikipedia.org/wiki/MySQL" \o "MySQL)-сервером версій 4.1.2 і вище, реалізовуючи не тільки процедурний, але і об'єктно-орієнтований інтерфейс до MySQL. Додаткові можливості цього модуля включають - [SSL](https://uk.wikipedia.org/wiki/SSL), [контроль транзакцій](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D0%B7%D0%B0%D0%BA%D1%86%D1%96%D1%8F_(%D0%B1%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)), [підтримка реплікації](https://uk.wikipedia.org/w/index.php?title=%D0%9F%D1%96%D0%B4%D1%82%D1%80%D0%B8%D0%BC%D0%BA%D0%B0_%D1%80%D0%B5%D0%BF%D0%BB%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%97&action=edit&redlink=1) та інші.

Шоста версія PHP розроблялася з жовтня 2006 року. Було зроблено безліч нововведень, як, наприклад, виключення з ядра регулярних виразів POSIX і «довгих» суперглобальних масивів, видалення директив safe\_mode, magic\_quotes\_gpc і register\_globals з конфігураційного файлу php.ini. Одним з основних нововведень повинна була стати підтримка Юнікоду. Однак у березні 2010 року розробка PHP6 була визнана безперспективною через складнощі з підтримкою Юнікоду. Вихідний код PHP6 переміщений на гілку, а основною лінією розробки стала версія 5.4.

У 2014 році було проведено голосування, за результатами якого наступна версія отримала назву PHP 7. Вихід нової версії планувався в середині жовтня 2015 року. У березні 2015 року Zend представили інфографіку в якій описані основні нововведення PHP 7. 3 грудня 2015 року було оголошено про вихід PHP версії 7.0.0. Нова версія грунтується на експериментальній гілці PHP, яка спочатку називалася phpng (PHP Next Generation - наступне покоління), і розроблялася з упором на збільшення продуктивності і зменшення споживання пам'яті. У новій версії додана можливість вказувати тип повертаються з функції даних, доданий контроль переданих типів для скалярних даних, а також нові оператори.

* 1. **Синтаксис мови**

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 у документах, які відповідають правилам [XML](https://uk.wikipedia.org/wiki/XML). Також можна використовувати скорочений запис <? ?> (інколи потрібно активізувати даний стиль у файлі налаштувань інтерпретатора php.ini: змінна short\_open\_tag повинна мати значення On) і записом у стилі [ASP](https://uk.wikipedia.org/wiki/Active_Server_Pages): <% %> (в php.ini змінна asp\_tags повинна мати значення On).

Основою будь-якого програмування є [змінні](https://uk.wikipedia.org/wiki/%D0%97%D0%BC%D1%96%D0%BD%D0%BD%D0%B0" \o "Змінна). PHP, як і деякі інші [UNIX](https://uk.wikipedia.org/wiki/UNIX)-[скриптові](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82" \o "Скрипт) мови, не потребує їх опису. Існує три стилі оформлення змінних у PHP:

* короткий стиль - змінні записуються у вигляді: $variable. Використовується в більшості випадків. Змінна, яка створюється програмним кодом, записується таким чином. Можливо також використання такого стилю для добування 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. Дані змінні називаються також глобальними.
* Довгий стиль - $HTTP\_POST\_VAR['variable'], $HTTP\_GET\_VARS['variable']. Найповніший стиль запису і найменш використовуваний. Починаючи з PHP 5, ці змінні можна вимкнути. Використовувати змінні такого типу не рекомендовано задля кращої сумісності із майбутніми версіями PHP.

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, зазвичай, розрядність становить 32 біти. PHP не підтримує беззнакові цілі числа. Дійсні числа із плаваючою комою можуть бути подані в десятковій або експоненційній формі. Для кожної змінної можна надати власний тип даних.

[Оператори](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) з погляду мови програмування дозволяють виконувати відповідну дію над одним чи кількома операндами. Оператори бувають трьох типів - [унарні](https://uk.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B0%D1%80%D0%BD%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F), [бінарні](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%BD%D0%B0%D1%80%D0%BD%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F) та [тернарні](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%80%D0%BD%D0%B0%D1%80%D0%BD%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F" \o "Тернарна операція). Оператори, як і в інших мовах, характеризуються не лише дією, а й [асоціативністю](https://uk.wikipedia.org/wiki/%D0%90%D1%81%D0%BE%D1%86%D1%96%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) та [пріоритетністю](https://uk.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D1%96%D0%BE%D1%80%D0%B8%D1%82%D0%B5%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C&action=edit&redlink=1). Особливістю [булевих операцій](https://uk.wikipedia.org/wiki/%D0%91%D1%83%D0%BB%D0%B5%D0%B2%D0%B0_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0" \o "Булева алгебра) порівняння - розрізнення двох класів - з урахуванням типу і без нього, при якому відбувається приведення до відповідного типу. Округлення відбувається завжди в менший бік. У мові реалізовані особливі класи операторів - виконання, управління помилками та перевірки належності до класу.

[Функції](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) з погляду мови програмування 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](https://uk.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).

У мові реалізована функціональність посилань. Можливо створити необмежену кількість 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. Натомість прийнятого в багатьох мовах принципу [перевантаження функцій](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%97), що дозволяє змінити хід виконання певної функції залежно від типу та кількості вхідних параметрів, у PHP використовується метод динамічних аргументів. Це дає змогу не визначати кількість параметрів функцій при їх оголошенні, а працювати з 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 оминути, 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](https://uk.wikipedia.org/wiki/%D0%97%D0%B1%D0%B8%D1%80%D0%B0%D1%87_%D1%81%D0%BC%D1%96%D1%82%D1%82%D1%8F). Проте за потреби можливе власноруч кероване виконання очищення пам'яті від надлишкових сегментів даних під час виконання скриптів, але використання функцій очищення пам'яті не є виправданим, 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 скрипту файли можуть повертати значення.

* 1. **Недоліки мови PHP**
* Незручність дизайну мови;
* Змінні з символом $;
* Складні назви поширених функцій (html\_entity\_decode,  тощо);
* Не підтримується [Unicode](https://uk.wikipedia.org/wiki/Unicode" \o "Unicode) у версіях до 6.0;
* Непередбачуваність нових версій PHP.