

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування

Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Статистичні моделі сценарного програмування засобами РНР-групи

Виконав: студент групи К-19
спеціальності 122 Комп'ютерні науки
Чарієв Овездари

Керівник: професор каф. АСМНСІ, д-р т. н.,
доцент Великодний Станіслав Сергійович

Консультант _____

Рецензент: канд. техн. наук, доцент
Фразе-Фразенко Олексій Олексійович

Одеса 2023

ЗМІСТ

Вступ.....	5
1 Огляд і аналіз предметної області	8
1.1 Історія виникнення мови PHP.....	8
1.2 Аналіз інформаційних технологій та засобів розробки програмного забезпечення	11
1.2.1 HTML	12
1.2.2 CSS.....	13
1.2.3 Flash	14
1.2.4 JavaScript	15
1.2.5 PHP	17
1.2.6 MySQL.....	17
1.2.7 Apache HTTP-сервер.....	19
1.2.8 AJAX.....	20
1.3 Аналіз редакторів початкового коду.....	22
1.3.1 AkelPad	22
1.3.2 Notepad++.....	23
1.3.3 PHP Expert Editor.....	25
1.3.4 Scriptomania	26
1.3.5 PSPad	26
1.3.6 DzSoft PHP Editor	28
2 Формування моделей об'єкта дослідження	30
2.1 Типи даних.....	31
2.2 Оператори в середовищі розробки PHP.....	44
3 Результати теоретичних та експериментальних досліджень.....	48
3.1 Налаштування конфігурації	48
3.2 Результати досліджень	57
Висновки	63
Перелік джерел посилання	65

ВСТУП

Головна область застосування PHP – написання скриптів, що працюють на стороні сервера; таким чином, мова PHP здатна виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти даних форм, генерувати динамічні сторінки або відсилати й приймати cookies, але мова PHP здатна виконувати набагато більше.

В області програмування для мережі Інтернет, мова PHP – одна з популярних середовищ скриптових мов (поряд з JSP, Perl і мовами, використовуваними в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності, багатоплатформеності і розповсюдженню початкових кодів на основі ліцензії PHP. Мова PHP має багато можливостей і саме тому ця тема є актуальною.

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних систем управління базами даних (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Інтерфейс PDO);
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з cookies і сесіями;
- робота з локальними і віддаленими файлами, сокетами;
- обробка файлів, що завантажуються на сервер;
- робота з XForms.

Таким чином, обираючи PHP, програміст отримує свободу вибору операційної системи і веб-сервера. Більш того, у нього з'являється вибір між використанням процедурного або об'єктно-орієнтованого програмування (ООП) або ж їх поєднання.

З використанням PHP розроблено безліч додатків, які широко використовуються на різних сайтах, форумах і блогах:

- Drupal, TYPO3, Mambo, Joomla, PHP-Nuke, Wordpress – системи управління вмістом (CMS);
- Magento, osCommerce – системи для Інтернет-комерції;
- Coppermine, Gallery Project, 4images – галереї зображень;
- MediaWiki, DokuWiki – вікі-движки;
- phpBB, SMF, vBulletin, Invision Power Board – форумні движки;
- phpMyAdmin, phpPgAdmin, Adminer – утіліти адміністрування СУБД;
- eyeOS – віддалена операційна система, заснована на принципі Desktop Operating System.

В силу популярності мови PHP і бажання збільшити швидкодію заснованих на ньому веб-додатків, створено декілька альтернативних компіляторів, близьких до PHP мови. Так в лютому 2010 року компанія Facebook відкрила свій компілятор PHP – HipHop (HPHP, Hyper-PHP) генерує код на C++, з наступною компіляцією в машинний код за допомогою gcc.

Отже, тема, що передбачає розробку та дослідження статистичних моделей сценарного програмування засобами PHP-групи – є актуальною.

Мета роботи полягає у всебічному аналізі засобів створення сценарних додатків.

Об'єктом роботи став синтез програмного коду мовою PHP.

Предметом роботи – сценарні додатки.

Для досягнення мети роботи, було встановлено наступний ряд завдань:

- а) виконано огляд предметної галузі;

б) проведено аналіз інформаційних технологій та засобів розробки програмного забезпечення;

в) проведено аналіз редакторів початкового коду;

г) сформульовано об'єкт моделі дослідження;

д) проведено аналіз віртуальних серверів;

е) розроблено засоби моделювання предметної галузі;

ж) розроблено технічне завдання;

и) розроблено фізичну модель програмного забезпечення.

Методи, що використовуються у роботі – об'єктно-орієнтоване програмування, процедурні методи, статичні методи (методи класів) тощо.

1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія виникнення мови PHP

Мова PHP була розроблена як інструмент для вирішення суто практичних завдань. Її творець, Расмус Лердорф, хотів знати, скільки людей читають його online-резюме, і написав для цього простеньку CGI-оболонку на мові Perl, тобто це був набір Perl-скриптів, призначених виключно для певної мети – збору статистики відвідувань [1].

Для довідки, CGI (Common Gateway Interface – загальний інтерфейс шлюзів) є стандартом, який призначений для створення серверних додатків, що працюють по протоколу HTTP. Такі додатки (їх називають шлюзами або CGI-програмами) запускаються сервером в режимі реального часу. Сервер передає запити користувача CGI-програмі, яка їх обробляє і повертає результат своєї роботи на екран користувача. Таким чином, відвідувач отримує динамічну інформацію, яка може змінюватися в результаті впливу різних факторів. Сам шлюз (скрипт CGI) може бути написаний на різних мовах програмування – C / C++, Fortran, Perl, TCL, UNIX Shell, Visual Basic, Python тощо.

Незабаром з'ясувалося, що оболонка володіє невеликою продуктивністю, і довелося переписати її заново, але вже на мові C. Після цього вихідники були викладені на загальний огляд для виправлення помилок та доповнення. Користувачі сервера, де розташовувався сайт з першою версією PHP, зацікавилися інструментом, з'явилися охочі його використовувати. Так що скоро PHP перетворився на самостійний проект, і на початку 1995 року вийшла перша відома версія продукту, називалася Personal Home Page Tools (засоби для персональної домашньої сторінки) [2]. Засоби ці були більш ніж скромними: аналізатор коду, що розуміє всього лише кілька спеціальних команд, і набір утиліт, корисних для створення гостьової книги, лічильника відвідувань, чату тощо.

До середини 1995 року після ґрунтовної переробки з'явилася друга версія продукту, названа PHP / FI (Personal Home Page / Forms Interpreter – персональна домашня сторінка / інтерпретатор форм). Вона включала набір базових можливостей сьогодишнього PHP, можливість автоматично обробляти html-форми і вбудовуватися в html-коди. Синтаксис PHP / FI сильно нагадував синтаксис Perl, але був більш простим [3].

У 1997 вийшла друга версія Сі-реалізації PHP – PHP / FI 2.0. До того моменту PHP використовували вже декілька тисяч людей по всьому світу, приблизно з 50 тис. доменів, що становило близько 1% всього числа доменів Internet. Число розробників PHP збільшилося до декількох чоловік, але, незважаючи на це, PHP / FI 2.0 все ще залишався великим проектом однієї людини. Офіційно PHP / FI 2.0 вийшов тільки в листопаді 1997 року, проіснувавши до цього в основному в бета-версіях. Незабаром після виходу його замінили альфа-версії PHP 3.0.

PHP 3.0 була першою версією, що нагадує PHP, яким ми знаємо його сьогодні. Вона дуже сильно відрізнявся від PHP / FI 2.0 і з'явився знову ж таки як інструмент для вирішення конкретної прикладної задачі. Його творці, Енді Гутманс (Andi Gutmans) і Зів Сураські (Zeev Suraski), в 1997 році переписали заново код PHP / FI, оскільки він здався їм непридатним для розробки додатку електронної комерції, над яким вони працювали. Для того щоб отримати допомогу в реалізації проекту від розробників PHP / FI, Гутманс і Сураські вирішили об'єднатися з ними і оголосити PHP3 офіційним наступником PHP / FI. Після об'єднання розробка PHP / FI була повністю припинена.

Однією з найсильніших сторін PHP 3.0 була можливість розширення ядра. Саме властивість розширюваності PHP 3.0 привернула увагу багатьох розробників, охочих додати свій модуль розширення. Крім того, PHP 3.0 надавала широкі можливості для взаємодії з базами даних, різними протоколами і API. Важливим кроком до успіху виявилася розробка нового, набагато потужнішого і повного синтаксису з підтримкою ООП. З моменту появи PHP 3.0 змінилася не тільки функціональність і внутрішня побудова мови, але й її

назва. В аббревіатурі PHP більше не було згадки про персональне використання, PHP стало скороченням (рекурсивним акронімом) від PHP: Hypertext Preprocessor, що значить «PHP: препроцесор гіпертексту».

До кінця 1998 року число користувачів PHP зросло до десятків тисяч. Сотні тисяч web-сайтів повідомляли про те, що вони працюють з використанням цієї мови. Майже на 10% серверів Internet була встановлена мова PHP 3.0.

Офіційно PHP 3.0 вийшла в червні 1998 року, після 9 місяців публічного тестування. А вже до зими Енді Гутманс і Зів Сураські почали переробку ядра PHP. У їх завдання входило збільшення продуктивності роботи складних додатків і поліпшення модульності коду, що лежить в основі PHP.

Нове ядро було названо «Zend Engine» (від імен творців: Zeev і Andi) і вперше представлено в середині 1999 року. Версія PHP 4.0, була заснована на цьому ядрі і принесла з собою набір додаткових функцій. Вона офіційно вийшла в травні 2000 року, майже через два роки після свого попередника, PHP 3.0. Крім поліпшення продуктивності, PHP 4.0 мала ще декілька ключових нововведень, таких як підтримка сесій, буферизація висновку, безпечніші способи обробки вводиться користувачем, і декілька нових мовних конструкцій.

П'ята версія PHP була випущена розробниками 13 липня 2004 року. Зміни включають оновлення ядра Zend (Zend Engine 2), що суттєво збільшило ефективність інтерпретатора. Введена підтримка мови розмітки XML. Повністю перероблені функції ООП, які стали багато в чому схожі з моделлю, яка використовується в Java. Зокрема, введени деструктор, відкриті, закриті і захищені члени й методи, остаточні члени й методи, інтерфейси та клонування об'єктів. Нововведення, проте, були зроблені з розрахунком зберегти найбільшу сумісність з кодом на попередніх версіях мови. На даний момент останньої стабільної версією є PHP 5.3, яка містить ряд змін та доповнень:

- збільшена швидкість роботи, приблизно на 10 – 20%;
- введено простір імен;
- пізні статичні зв'язування і спеціальний метод `callStatic()`;
- лямбда-функції та замикання;

- додавання розширень: intl, phar, fileinfo і sqlite3;
- опціональний збирач сміття;
- був написаний драйвер MySQLnd для самої популярної у PHP розробників бази даних Mysql. З появою нативного драйвера швидкість роботи з Mysql значно збільшилася, також новий драйвер доступний вже в стандартній зборці (раніше це було неможливо через ліцензії, пов'язаної з libmysql);
 - скасування підтримки версій операційної системи до Windows 2000 (Windows 98, NT4, тощо);
 - нові синтаксичні конструкції, такі як NOWDOC, обмежений GOTO, короткий вид тернарного оператора «?:» тощо.

Шоста версія PHP перебувала в стадії розробки з жовтня 2006 року. У ній вже були зроблені 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`. Основні зусилля були зосереджені на підтримці Юнікоду. Однак у березні 2010 року розробка PHP 6 була визнана безперспективною через складнощі з підтримкою Юнікоду. Вихідний код PHP 6 переміщений на гілку загального доступу, а основною лінією розробки стала версія 5.4.

1.2 Аналіз інформаційних технологій та засобів розробки програмного забезпечення

Для створення сучасних сайтів зазвичай використовуються: HTML, CSS, JavaScript (або VBScript), FLASH, PHP (або Perl) і реляційна база даних MySQL. Так само потрібен веб сервер Apache для установки сайту.

Клієнтське середовище (браузер, наприклад Firefox) є переднім краєм роботи програми. У цьому середовищі відображаються HTML-сторінки у вікні та обслуговуються історії сеансів HTML-сторінок, що відображаються у браузері протягом сесії. Об'єкти цього середовища, отже, повинні мати мож-

ливість маніпулювати сторінками, вікнами та історією. Для цього потрібна клієнтська мова – PHP (або Perl) та реляційна база даних MySQL.

1.2.1 HTML

HTML – це не мова програмування і не мова оформлення документів. Це, в першу чергу, засіб розмітки тексту.

Мова HTML містить достатню кількість елементів, що дозволяють оформити документ на будь-яку тематику. Якщо не влаштовує спосіб оформлення документа браузером (потрібно змінити шрифт, запропонований за умовчанням, або зробити його подрібніше) – додавши HTML-теги форматування тексту можна добитися потрібних результатів. Якщо потрібно більш детально налаштувати зовнішній вигляд документа, можна скористатися таблицями стилів (CSS).

Динамічний HTML або DHTML – це комерційний термін, придуманий для опису технологій, які були введені в четвертій версії Web-браузерів і дозволяли обходити обмеження HTML. DHTML являє собою комбінацію Web-стандартів CSS + JavaScript + DOM + XHTML = DHTML де:

- CSS – визначає атрибути об'єктів;
- JavaScript – змінює об'єкти;
- DOM – знаходить об'єкти;
- XHTML – створює об'єкти (виконує розмітку тексту).

Створена на основі DHTML сторінка може змінюватися без звернення до сервера за додатковими даними, тобто являють собою клієнтський код (client-side-code), який в свою чергу, дозволяє істотно збільшити інтерактивність Web-сторінок. Розглянемо переваги та недоліки DHTML у табл. 1.1.

Таблиця 1.1 – Переваги та недоліки DHTML

Переваги	Недоліки
<p>Підтримується всіма браузерами.</p> <p>Використовує стандартні технології.</p> <p>Можна вносити зміни в Web-сторінку після її завантаження.</p> <p>Текстові файли DHTML завантажуються швидше, ніж Flash і Java.</p> <p>Не вимагає модулів розширення.</p> <p>Висока швидкість розробки Web-сторінок.</p>	<p>Різний вид Web-сторінок через несумісність браузерів і операційних систем.</p> <p>JavaScript і CSS вельми чутливі до помилок в синтаксисі.</p> <p>Ненадійна робота у зв'язку з помилками в браузерах.</p>

1.2.2 CSS

CSS (Cascading Style Sheets) – мова таблиць каскадних стилів. Він розроблений для того, щоб розширити можливості з оформлення Web-сторінок.

CSS використовується веб-розробниками для завдання зовнішнього вигляду (шрифтів, кольорів, відступів, розташування та ін) веб-сторінок. CSS розроблений для відділення основного вмісту документа (написаного на мові розмітки, наприклад HTML) від оформлення цього вмісту (написаного на CSS) [4]. Таке відділення надає веб-розробникам більшу гнучкість, спрощує завдання зовнішнього вигляду документів та оформлення повторюваних елементів розмітки.

При використанні HTML і CSS важливо розуміти наступне:

- HTML-код формує текст логічно, тобто задає структури Web-сторінки: розташування і порядок проходження абзаців, графічних зображень, рядків і осередків в таблиці і особливе значення окремих фрагментів тексту.

- таблиці стилів CSS формують тексти фізично, тобто задають подання Web-сторінки: яким шрифтом будуть набрані звичайний текст абзаців, яким кольором виділити заголовки, чи будуть у таблиці рамка тощо.

Правила гарного тону веб-дизайну [5] вимагають, щоб подання веб-сторінки було відокремлено від її структури. Тому по можливості потрібно виносити визначення стилів CSS в окремі файли (таблиці стилів).

Каскадні таблиці стилів по суті своїй не динамічні. Вони дозволяють визначати, як буде виглядати документ при завантаженні і не більше того. Але властивості веб-сторінок, створених за допомогою CSS, можна динамічно змінювати за допомогою мови JavaScript [6].

1.2.3 Flash

Adobe Flash (раніше Macromedia Flash), або просто Flash – мультимедійна платформа компанії Adobe для створення веб-додатків та мультимедійних презентацій. Широко використовується для створення рекламних банерів, анімації, ігор, а також відтворення на веб-сторінках відео-і аудіозаписів.

Стандартним розширенням для скопільованих flash-файлів (анімації, ігор та інтерактивних додатків) є. SWF (Shockwave Flash або Small Web Format). Відеоролики в форматі Flash представляють собою файли з розширенням FLV (при цьому Flash в даному випадку використовується тільки як контейнер для відеозапису). Розширення FLA відповідає формату робочих файлів в середовищі розробки. Розглянемо переваги та недоліки Flash у наступній таблиці [7].

Таблиця 1.2 – Переваги та недоліки Flash

Переваги	Недоліки
Платформна незалежність, тобто сторінки створені за допомогою Flash виглядають однаково в різних операційних системах і різних браузерях. Поширеність та надання дизайнерам величезна кількість інкрементів.	Flash-технологія складна у вивченні. Не бажання багатьох користувачів Інтернет оновлювати версії програвачів Flash, і отже багато нових Flash-модулів не працюють. Проблеми з інтерфейсом і його ста-

Правильно створені Flash-файли займають мало місця і досить швидко завантажуються.	ндартизацією. Довга завантаження через великий обсяг графічних файлів.
--	---

Дизайнерам веб-сайтів часто доводиться робити вибір між Flash і DHTML. Якщо DHTML дозволяє працювати з інтерактивністю за допомогою HTML, CSS і JavaScript, то Flash являє собою деякий об'єкт, що вбудовується в HTML-сторінку, і є окремою Internet-технологією.

1.2.4 JavaScript

Популярність мови JavaScript пов'язана з його широкими можливостями по взаємодії з елементами веб-сторінки без її перезавантаження. Це дозволяє ховати і показувати фрагменти дизайну, переміщати їх і змінювати оформлення. Шляхом таких дій можна створювати презентаційні ефекти, меню, невеликі ігри, обробляти дані форм і керувати вмістом. Розглянемо деякі можливості мови JavaScript:

1) Підтримка верствів. Верстви надають собою фрагменти HTML, які можна розміщувати на веб-сторінці шляхом накладення їх один на одного з точністю до пікселя. Верстви все більше набувають визнання як засіб верстки веб-сторінок і створення різних ефектів. Частково це пов'язано з тим, що їх параметри легко змінюються динамічно через скрипти, що дає можливість створювати меню та інші виразні засоби дизайну.

2) Робота з формами. Через скрипти зручно отримувати та обробляти будь-які дані форм, це дозволяє перевірити інформацію на правильність введення перед її відправкою на сервер. Можна створити «захист від дурня» для контролю того, щоб в елементи форм вводилися коректні повідомлення. Так, якщо в текстовому полі просять ввести число, слід відразу припинити можливість введення нечислових символів.

3) Робота із зображеннями. Через скрипти можна робити попередню завантаження зображень. Ще до прямого звернення до зображень, браузер поміщає їх в свою пам'ять, щоб по необхідності швидше відобразити в документі. Основні параметри зображень, такі як: ширина, висота зображень і адреса графічного файлу, теж можна міняти динамічно. Це дозволяє створювати ефект перекочування, коли малюнок змінюється на інший при наведенні на нього курсора миші й назад, коли курсор виводиться з зображення. Подібним методом робляться і галереї фотографій – зміна зображень відбувається в межах однієї сторінки [8].

4) Робота з cookies. JavaScript підтримує повноцінну роботу з cookies – невеликі текстові файли на локальному комп'ютері, в яких зберігається технічна інформація. Cookies можна використовувати для збереження дати останнього відвідування читача, паролів, а також будь-якої інформації про дії відвідувача на сайті. Подібне застосування дозволяє персоналізувати сайт і зробити його більш зручним для відвідувачів.

5) Відстеження подій. Подією називається певну дію користувача або зміна стану документа. JavaScript відстежує більшість подій і дозволяє визначати реакцію на них. Наприклад, при завантаженні веб-сторінки відбувається подія onLoad. Якщо необхідно запустити скрипт відразу після завантаження документа, слід цієї події призначити функцію, яка буде виконуватися при його настанні.

6) Створення нових вікон. Скрипти дозволяють створювати нові вікна, задавати у цих вікон вид відображення і формувати їх вміст. Причому, у створених таким способом вікон можна відключати адресний рядок, смуги прокрутки, меню і управляти їх розміром та положенням на екрані.

7) Перевірка браузера. Підхід різних браузерів до об'єктної моделі, хоч і стандартизований, але ще не всіма до кінця прийнятий. Тому для створення універсального документа, однаково працює скрізь, доводиться робити перевірку, якою в даний момент використовується браузер, і давати йому відповідний код.

Зрозуміло, тут описані не всі можливості мови JavaScript, але і зазначеного достатньо, щоб зрозуміти, що це потужне, гнучкий, універсальний і зручний засіб розширити можливості веб-сторінок.

1.2.5 PHP

PHP – це мова програмування, призначена для створення сайтів. PHP дозволяє автоматизувати роботу з сайтом. Це скриптова-мова, що вбудовується в HTML, яка інтерпретується і виконується на сервері. Відмінність PHP від JavaScript, полягає в тому, що PHP-скрипт виконується на сервері, а клієнту передається результат роботи, тоді як в JavaScript-код повністю передається на клієнтську машину і тільки там виконується.

PHP є запроваджуваною мовою сценаріїв [9]. Мета мови полягає в тому, щоб дозволити веб-розробникам швидко створювати динамічно генеруються сторінки. Крім того, розробникам веб-додатків немає необхідності говорити, що веб-сторінки – це не тільки текст і картинки. Гідний уваги сайт повинен підтримувати деякий рівень інтерактивності з користувачем: пошук інформації, продаж продуктів, конференції тощо. PHP працює як частина веб-сервера. У цій мові немає строгої типізації даних і немає необхідності в діях по виділенню / звільненню пам'яті. Написаний PHP – код легко зрительно прочитати і зрозуміти.

За свою ідеологією PHP споконвічно був орієнтований на написанні невеликих скриптів. PHP це інтерпретована мова, і, внаслідок цього, не може зрівнятися по швидкості компіляції з С. Однак при написанні невеликих програм, що, загалом-то, властиво проектам на PHP, йому просто немає рівних.

1.2.6 MySQL

MySQL – це швидка і потужна система керування базами реляційними даних [10]. Вона дозволяє ефективно зберігати, шукати, сортувати і вибирати інформацію. Робота з базою даних має як переваги, так і недоліки.

Таблиця 1.3 – Переваги та недоліки MySQL

Переваги	Недоліки
<p>Значне скорочення коду (іноді в 2-3 рази) в порівнянні з файловими варіантами веб-додатків, що скорочує час розробки та спрощує процес налагодження.</p> <p>Висока швидкість виконання процедур запитів в СУБД, написані компільованому мовою С, яка на два порядки продуктивніше інтерпретується мови РНР.</p>	<p>Залежність програми не тільки від працездатності веб-сервера, але і від працездатності сервера даних. Ймовірність того, що один з двох серверів вийде з ладу вище, ніж при використанні одного.</p> <p>Робота з файловою системою здійснюється швидше, ніж при використанні СУБД. Так, пошукова система Google, відома високою швидкістю обробки запитів, заснована на моделі плоских файлів і не використовує бази даних.</p>

Застосування СУБД викликано в основному через те, що інформація в Інтернет може відображатися як пасивно (тобто користувач може тільки зчитувати її), так і активно – тоді користувач може додавати інформацію і редагувати її. Це: гостьові книги, форуми, чати, блоги, wiki-проекти, системи управління контентом і т.п.

Переваги використання СУБД в цьому випадку очевидні: всі рутинні операції по запису, пошуку і вилученню інформації бере на себе сервер SQL, програмісту залишається тільки зосередитися на логіці веб-додатки. MySQL управляє доступом до даних, дозволяючи працювати з ними одночасно декільком користувачам, при цьому гарантуючи надання доступу тільки тим користувачам, які мають на це право. СУБД стала стандартом для використання в веб-додатках завдяки своїй продуктивності, простоті установки і обслуговуванню.

1.2.7 Apache HTTP-сервер

Apache – це вільний веб-сервер, який приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Веб-сервери – основа Всесвітньої павутини.

Apache є кросплатформним ПО, підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основними достоїнствами Apache вважаються надійність і гнучкість конфігурації. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СУБД для аутентифікації користувачів, модифікувати повідомлення про помилки і т. д. Підтримує IPv6.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів. Теоретично, ядро apache може функціонувати в чистому вигляді, без використання модулів. Однак, функціональність такого рішення вкрай обмежена. Ядро Apache повністю написано на мові програмування C.

Apache має вбудований механізм віртуальних хостів. Він дозволяє повноцінно обслуговувати на одному IP-адресі безліч сайтів (доменних імен), відображаючи для кожного з них власне вміст. Для кожного віртуального хоста можна вказати власні настройки ядра і модулів, обмежити доступ до всього сайту або окремих файлів. Деякі МРМ, наприклад Apache-ІТК дозволяють запускати процес httpd для кожного віртуального хоста з окремими ідентифікаторами uid і guid. Також, існують модулі, що дозволяють враховувати і обмежувати ресурси сервера (CPU, RAM, трафік) для кожного віртуального хоста.

1.2.8 AJAX

Зараз веб-проекти стають все складніше і відповідно повільніше відбувається їх виконання, що в деяких випадках значно дратує користувачів. Та й розробка займає тривалий час. Рішення цієї проблеми лежить в двох напрямках:

1) Шаблони. В даний час розробка веб додатків прагне до розмежування клієнтської частини і серверної, цим і обумовлюється повсюдне використання шаблонів. Переплітати між собою різні технології ставати занадто дорого для часу розробника. Так, наприклад, всі стилі форматування виносяться в CSS або в XSL файли, HTML або XML дані зберігаються в інших розділах, серверні оброблювачі в третіх, бази даних в четвертих. І якщо ще 5 – 6 років тому практично скрізь можна було побачити переплетіння всього цього в одному файлі, то зараз це все частіше ставати рідкістю. При розробці більш складних проектів виникає необхідність в структурованості і легкості читання коду. Не слід засмічувати код програміста кодом верстальника, а код верстальника – правками дизайнера, і так далі. Виникає необхідність в розмежуванні роботи цих фахівців. Так, наприклад, дизайнер буде робити свою роботу, верстальник свою, програміст свою, і при цьому ніхто один одному заважати не буде. У підсумку кожному учаснику проекту достатньо буде знати тільки ті дані, з якими йому доведеться працювати. В такому випадку продуктивність групи і якість проекту підвищується в рази.

2) AJAX – це радикально нова технологія. Спробуйте уявити JavaScript, що володіє всім можливостями PHP або Perl, включаючи роботу з графікою та базами даних, яка має набагато більш зручну розширюваність і практичність.

AJAX (Asynchronous Javascript and XML – «асинхронний JavaScript і XML») – це підхід до побудови інтерактивних користувацьких інтерфейсів веб-додатків, що полягає в «фоновому» обміні даними браузера з веб-сервером. В результаті при оновленні даних веб-сторінка не перезавантажується повністю, і веб-додатки стають швидшими і зручнішими. AJAX – один з компонентів концепції DHTML.

Переваги використання:

- економія трафіку. Використання AJAX дозволяє значно скоротити трафік при роботі з веб-додатком завдяки тому, що часто замість завантаження всієї сторінки достатньо завантажити тільки невелику змінилася частина.
- зменшення навантаження на сервер. AJAX дозволяє дещо понизити навантаження на сервер. Наприклад, в Gmail, коли ви відзначаєте прочитані листи, серверу достатньо внести зміни в базу даних і відправити клієнтському скрипту повідомлення про успішне виконання операції без необхідності повторно створювати сторінку і передавати її клієнту.
- прискорення реакції інтерфейсу. Оскільки потрібно завантажити тільки змінилася частина, то користувач бачить результат своїх дій швидше.
- недоліки використання:
 - інтеграція із стандартними інструментами браузера. Динамічно створювані сторінки не реєструються браузером в історії відвідин сторінок, тому не працює кнопка «Назад» надає користувачам можливість повернутися до переглянутих раніше сторінок, але існують скрипти, які можуть вирішити цю проблему. Інший недолік зміни вмісту сторінки при постійному URL полягає в неможливості збереження закладки на бажаний матеріал. Частково вирішити ці проблеми можна за допомогою динамічної зміни ідентифікатора фрагмента (частини URL після #), що дозволяє чимало браузерів.
 - динамічно завантажувати вміст недоступний пошуковикам. Пошукові машини не можуть виконувати JavaScript, тому розробники повинні поклопотатися про альтернативні способи доступу до вмісту сайту.
 - старі методи обліку статистики сайтів стають неактуальними. Багато сервісів статистики ведуть облік переглядів нових сторінок сайту, для сайтів сторінки яких широко використовують AJAX, така статистика втрачає актуальність.

1.3 Аналіз редакторів початкового коду

Редактор початкового коду – текстовий редактор для створення і редагування початкового коду програм. Він може бути окремим додатком, або вбудований в інтегроване середовище розробки (IDE).

Редактори початкового коду мають деякі можливості, що спрощують і прискорюють написання і зміну коду, такі як підсвічування синтаксису, автодоповнення, перевірка правильності розстановки дужок, контекстна допомога по коду та багато інших. Такі редактори надають зручний засіб для запуску компілятора, інтерпретатора, відладчика або інших програм необхідних в процесі розробки програмного забезпечення. Незважаючи на те, що багато текстових редакторів можуть бути використані для редагування початкового коду, якщо вони не мають розширених можливостей, що автоматизують або спрощують введення і модифікацію коду, то вони не можуть називатися «редакторами початкового коду», а просто є «текстовими редакторами, які також можуть бути використані для редагування початкового коду».

1.3.1 AkelPad

AkelPad – вільний текстовий редактор з відкритим вихідним кодом для операційних систем Microsoft Windows, але може вільно бути запущений під Wine і працювати під управлінням Unix-подібних операційних систем, таких як Linux. Розповсюджується під ліцензією BSD [11].

До можливостей можна віднести: одновіконний режим (SDI), багатовіконний режим (MDI), псевдо-багатовіконний режим (PMDI); редагування файлів розміром більш 64К; повна підтримка Unicode рядків на Unicode системах (NT/2000/XP/2003); роботу з кодуваннями Unicode (UTF-16 little endian, UTF-16 big endian, UTF-8, UTF32-LE, UTF32-BE); роботу з будь-якою кодовою сторінкою, встановленою в системі, робота з DOS / Windows, Unix і Mac форматами перекладу рядка; попередній перегляд відкриття файлів, ко-

ректне відображення псевдографіки, блокове виділення тексту, багаторівневий відкат дій; швидкий пошук / заміна рядків тексту; запам'ятовування кодування і позиції курсора в файлі; друк (в тому числі кольоровий) і попередній перегляд, підтримка мовних модулів; підтримка плагінів (підсвічування синтаксису, автодоповнення, згортання коду запуск скриптів, клавіатурні макроси, можливість створювати власні меню, в тому числі контекстні, колірні теми і тощо) (рис. 1.1).

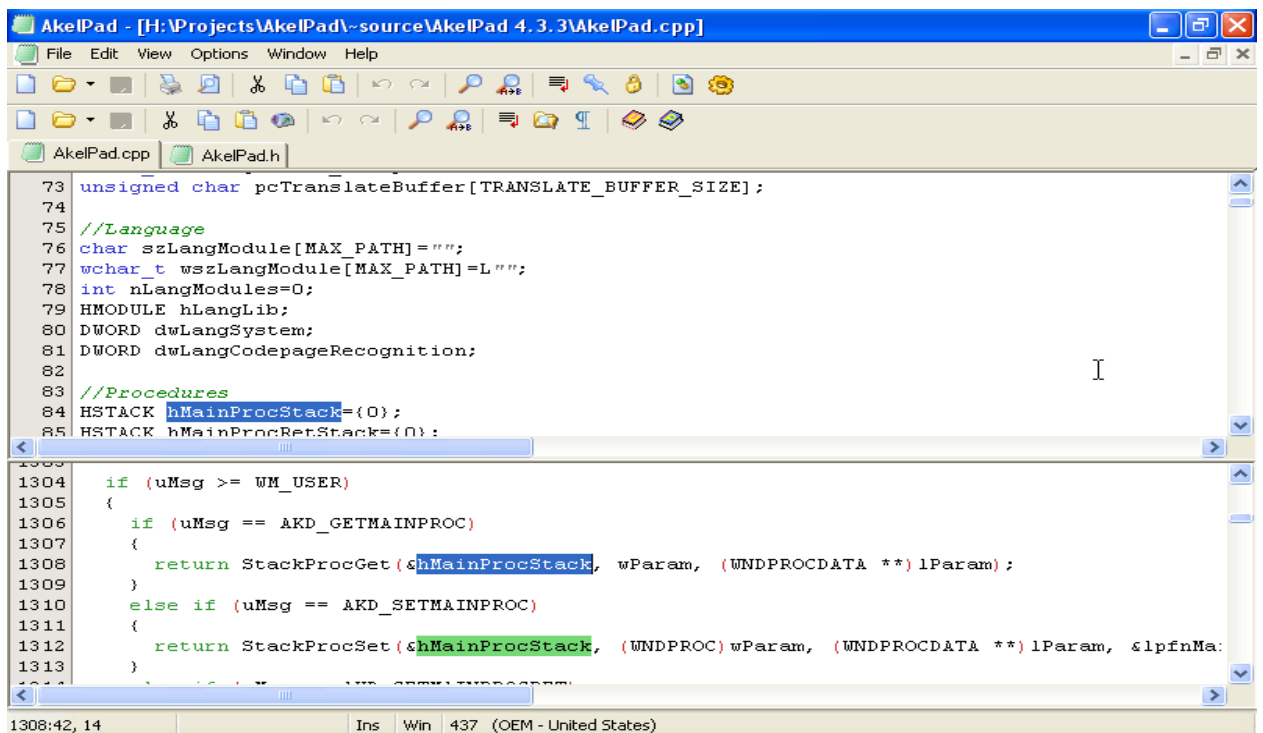


Рисунок 1.1 – Вид головного вікна AkelPad

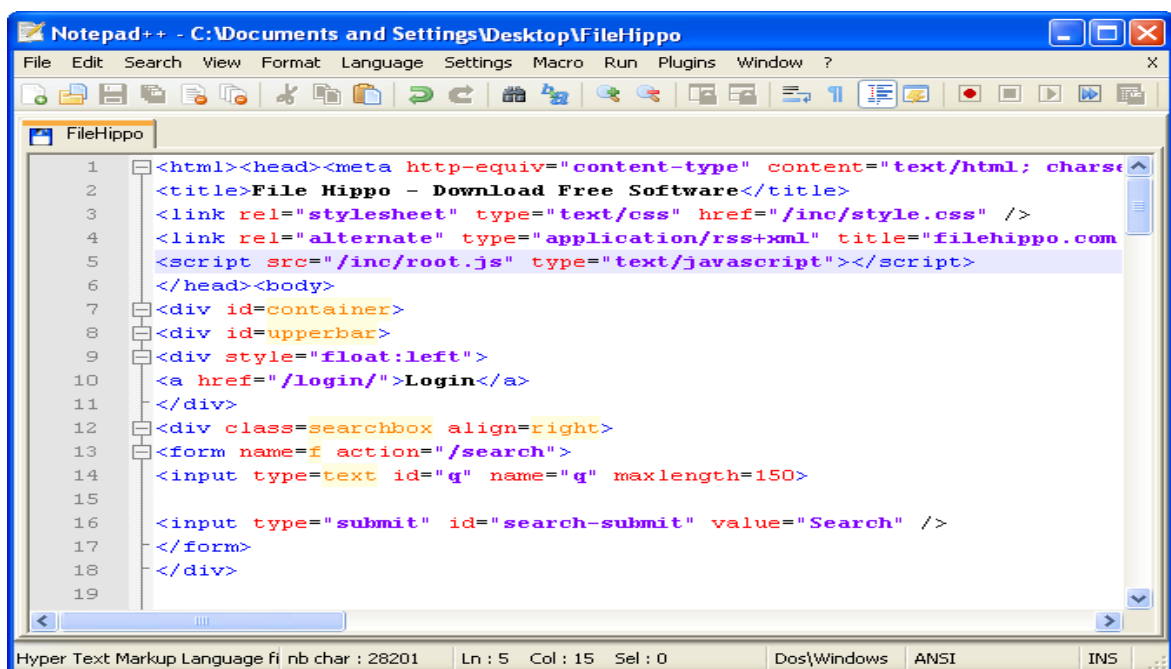
1.3.2 Notepad++

Notepad++ – вільний текстовий редактор для Windows з підсвічуванням синтаксису великої кількості мов програмування і розмітки. Він базується на компоненті Scintilla, написаний на C++ з використанням STL, а також Win32 API і поширюється під ліцензією GPL. Базова функціональність програми може бути розширена як за рахунок плагінів, так і сторонніх модулів, таких як компілятори і препроцесора [12].

До можливостей можна віднести підсвічування синтаксису, згортання коду, автодоповнення, закладки, регулярні вирази для пошуку і заміни, запис і відтворення макросів, порівняння файлів, перевизначення будь-яких гарячих клавіш, резервне копіювання зберігаються файлів (включається в налаштуваннях), трансформація тексту за допомогою підключеного плагіна TextFX, підтримка і конвертація кодувань ANSI, UTF-8 і UCS-2, блокове виділення тексту, одночасне виділення кількох різних місць (з CTRL).

При встановленні додаткових плагінів можна отримати можливість доступу до FTP-менеджеру (плагіни: NppFTP), HEX-редактор, авто-збереження (при втраті фокуса; через налаштовується проміжок часу), перевірка орфографії (з використанням GNU Aspell), потужне симетричне і асиметричне шифрування тексту (при установці плагіна NppDarkCrypt), підтримка Zen Coding.

Програма поставляється в двох версіях: UNICODE і ANSI, причому останній варіант доступний тільки при ручному розпакуванні архіву. Модулі можуть бути написані під певну версію програми, але частіше плагін працює в обох версіях, хоча останнім часом акцент робиться на UNICODE-версію, як найбільш перспективну (рис. 1.2).



The image shows a screenshot of the Notepad++ application window. The title bar reads "Notepad++ - C:\Documents and Settings\Desktop\FileHippo". The menu bar includes "File", "Edit", "Search", "View", "Format", "Language", "Settings", "Macro", "Run", "Plugins", and "Window". The toolbar contains various icons for file operations and editing. The main text area displays HTML code for a page titled "File Hippo - Download Free Software". The code includes a meta tag for content type, a title tag, links for stylesheets and RSS, a script tag for root.js, and a body section with a container, upper bar, login link, and a search form. The status bar at the bottom shows "Hyper Text Markup Language fi nb char : 28201", "Ln : 5 Col : 15 Sel : 0", "Dos\Windows", "ANSI", and "INS".

```
1 <html><head><meta http-equiv="content-type" content="text/html; charse
2 <title>File Hippo - Download Free Software</title>
3 <link rel="stylesheet" type="text/css" href="/inc/style.css" />
4 <link rel="alternate" type="application/rss+xml" title="filehippo.com
5 <script src="/inc/root.js" type="text/javascript"></script>
6 </head><body>
7 <div id=container>
8 <div id=upperbar>
9 <div style="float:left">
10 <a href="/login/">Login</a>
11 </div>
12 <div class=searchbox align=right>
13 <form name=f action="/search">
14 <input type=text id="q" name="q" maxlength=150>
15
16 <input type="submit" id="search-submit" value="Search" />
17 </form>
18 </div>
19
```

Рисунок 1.2 – Вид головного вікна Notepad++

1.3.3 PHP Expert Editor

PHP Expert Editor – редактор для PHP, Perl, Python, HTML, JavaScript та інших файлів з підтримкою UTF-8. Програма розроблена спеціально для PHP-розробників, в неї інтегрована клієнтська частина відладчика PHP DBG. Програма має вбудований HTTP-сервер і дозволяє запускати на стороні сервера скрипти на PHP, Perl, Python [13].

Перевірка синтаксису PHP, вбудований браузер, FTP-клієнт з підтримкою SFTP, оглядач коду і файлів, підтримка проектів, що настроюються шаблони коду, що настроюється підсвічування коду, і багато інших функцій для підвищення зручності розробки [14].

Функціональні можливості: підтримка UTF-8; налаштування підсвічування синтаксису, згортання коду; вбудовані веб-сервер і браузер, налагоджувальний запуск серверних скриптів; вбудований FTP-клієнт з підтримкою SFTP; оглядачі проекту, бібліотек, файлів, коду; налаштування гарячих клавіш, макроси, шаблони коду, автозбереження; підтримка мови PHP (підказка параметрів, контекстна довідка); підтримка Perl, Python, Ruby, Tcl (рис. 1.3).

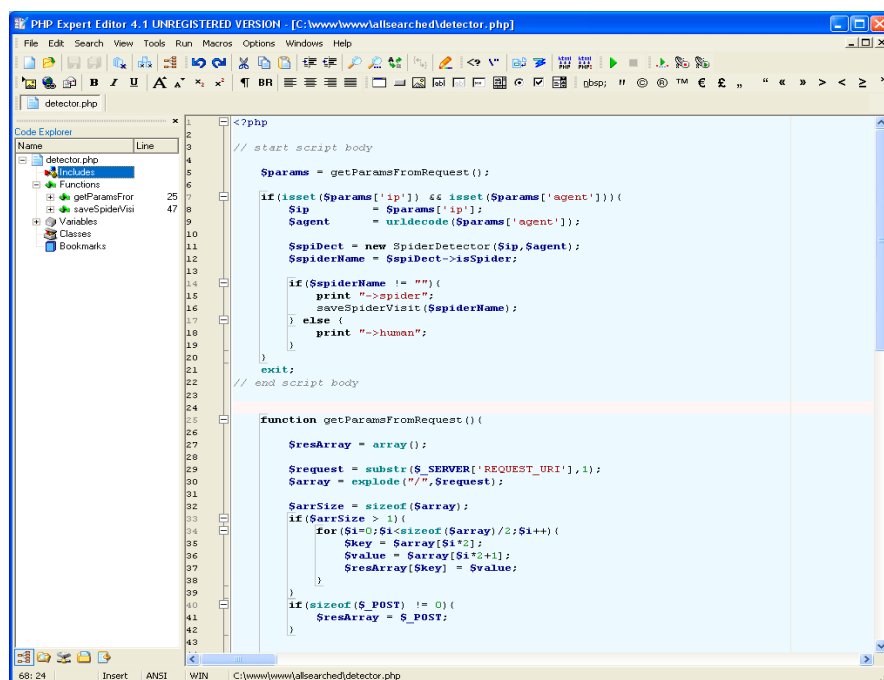


Рисунок 1.3 – Вид головного вікна PHP Expert Editor

1.3.4 Scriptomania

Красивий і зручний редактор PHP коду, що володіє високою функціональністю [15]. Основні особливості: редактор HTML, PHP, JavaScript, MySQL коду з просунутою системою підсвічування синтаксису, підтримка SQL макросів (може навіть робити запис дій миші), механізми автозавершення коду (HTML / PHP) + механізм корекції коду, можливість локального тестування скриптів, великі інструментальні засоби і багато іншого (див. рис 1.4).

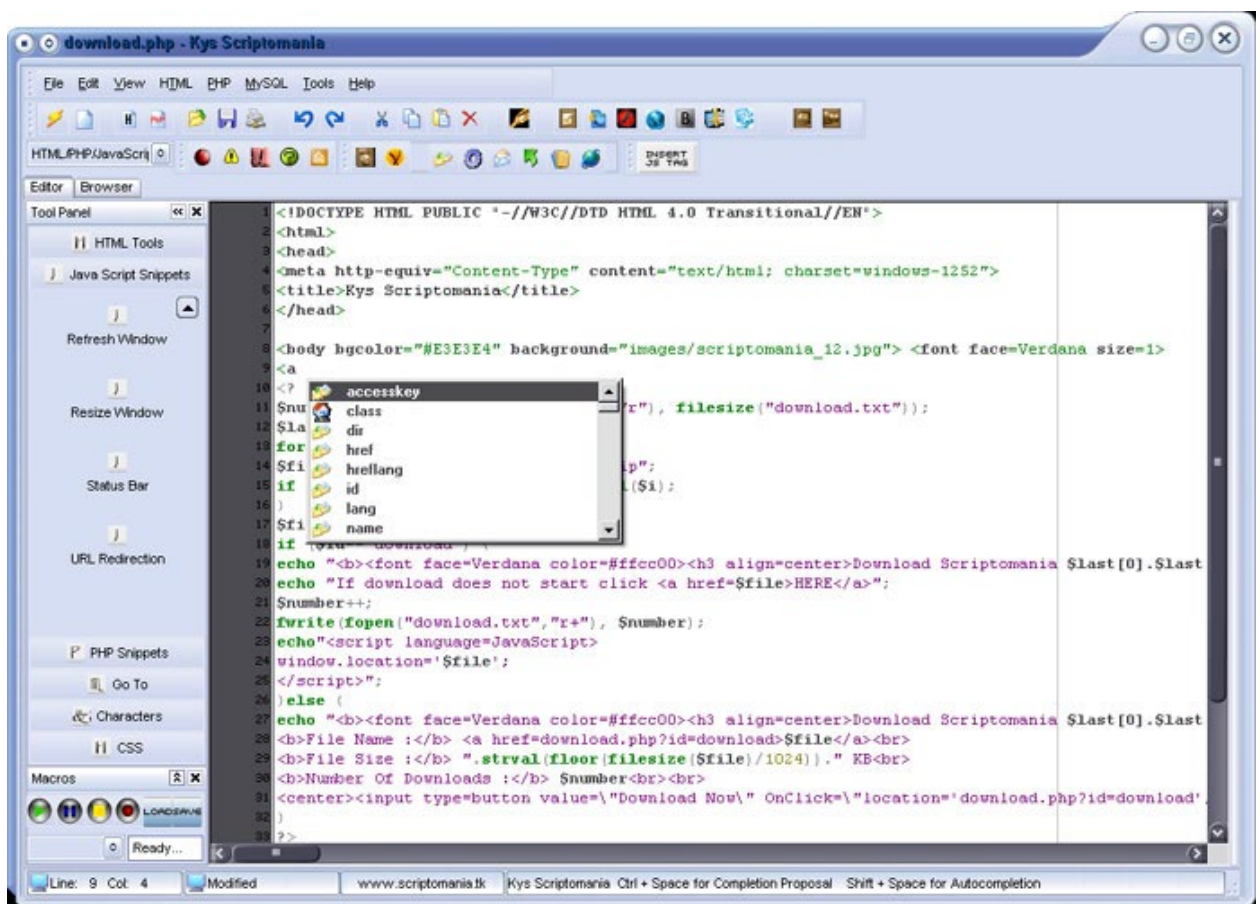


Рисунок 1.4 – Вид головного вікна Scriptomania

1.3.5 PSPad

PSPad – безкоштовний текстовий редактор і редактор початкових текстів програм для операційної системи Windows [16].

Можливості цього текстового редактора:

- робота з проектами;
- робота над декількома документами одночасно (MDI);
- збереження екранної сесії (при наступному вході PSPad автоматично відкриє всі відкриті на момент закриття файли);
- FTP клієнт - ви можете редагувати файли прямо з WEB-сервера;
- підтримка макросів записувати, зберігати і завантажувати макроси;
- пошук і заміна у файлах;
- порівняння текстів з різнокольоровою підсвіткою відмінностей;
- шаблони (HTML-теги, скрипти, шаблони коду);
- інсталяція містить шаблони для HTML, PHP, Pascal, JScript, VBScript, MySQL, MS-DOS, Perl [17].
- підсвічування синтаксису застосовується автоматично згідно типу файлу;
- автокорекція;
- інтелектуальний вбудований HTML-попередній перегляд використовуючи IE і Mozilla;
- повноцінний HEX редактор
- інтегрована TiDu-бібліотека для форматування та перевірки HTML-коду, конверсії в CSS, XML, XHTML;
- вБудована вільна версія CSS-редактора TopStyle Lite;
- експорт коду з підсвічуванням в форматах RTF, HTML, TeX в файл або буфер обміну;
- вертикальне виділення, закладки, мітки, нумерація рядків, ...
- Переформатування і стиснення HTML-коду, зміни регістра слів, тегів, букв;
- таблиця ASCII-символів з приведенням відповідності HTML-мнемонік;
- навігатор коду для Pascal, INI, HTML, XML, PHP, і багатьох інших у майбутньому (рис. 1.5);
- перевірка правопису;

- вбудований веб-браузер з підтримкою Apache;
- підсвічування парних дужок.

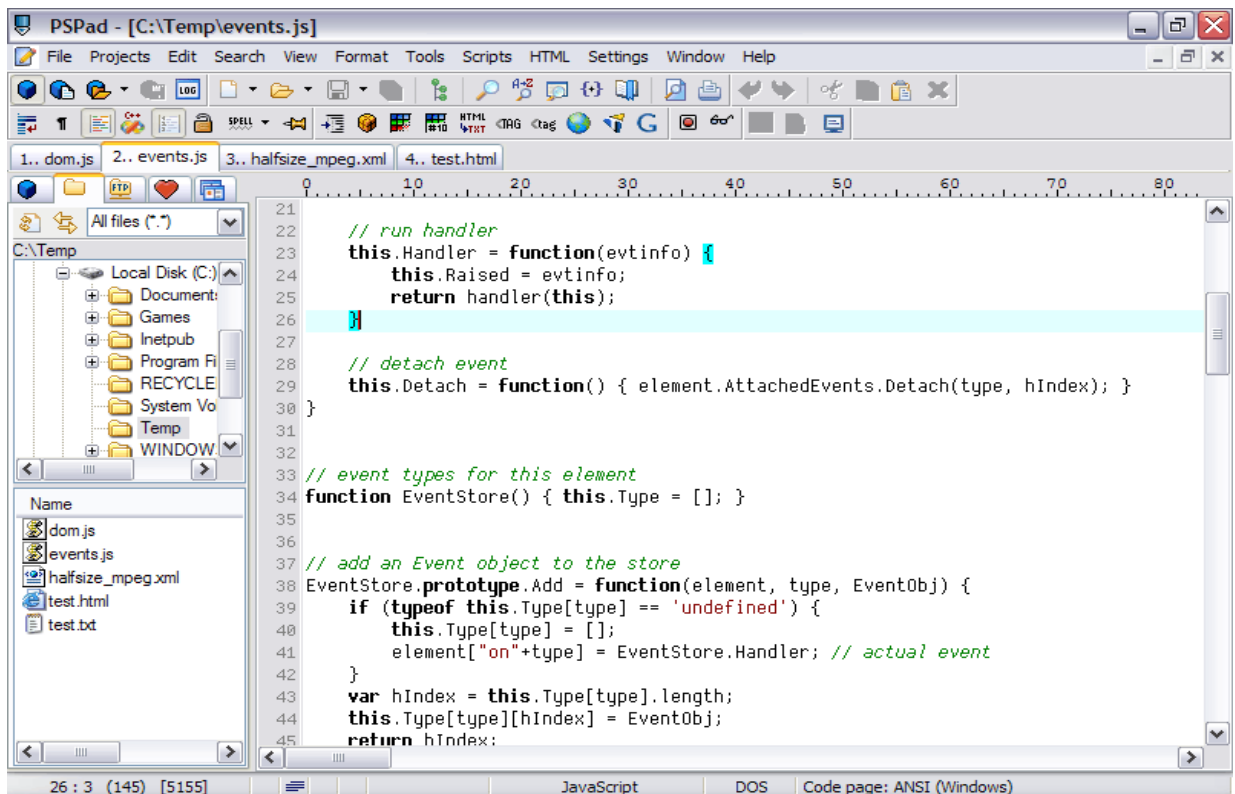


Рисунок 1.5 – Вид головного вікна PSPad

1.3.6 DzSoft PHP Editor

DzSoft PHP Editor – зручний і потужний інструмент для того, щоб писати і тестувати сценарії PHP і HTML / XML сторінки. Будучи розробленим визначено для PHP розробки, він має дружній, але потужний інтерфейс, який є зручним і для новачків, і для досвідчених програмістів [18]. Має зручний попередній перегляд у внутрішньому браузері (не потрібен зовнішній веб-сервер), швидкі вставки коду і багато інші необхідні особливості, які допоможуть у зручній і продуктивній розробки PHP скриптів (рис. 1.6).

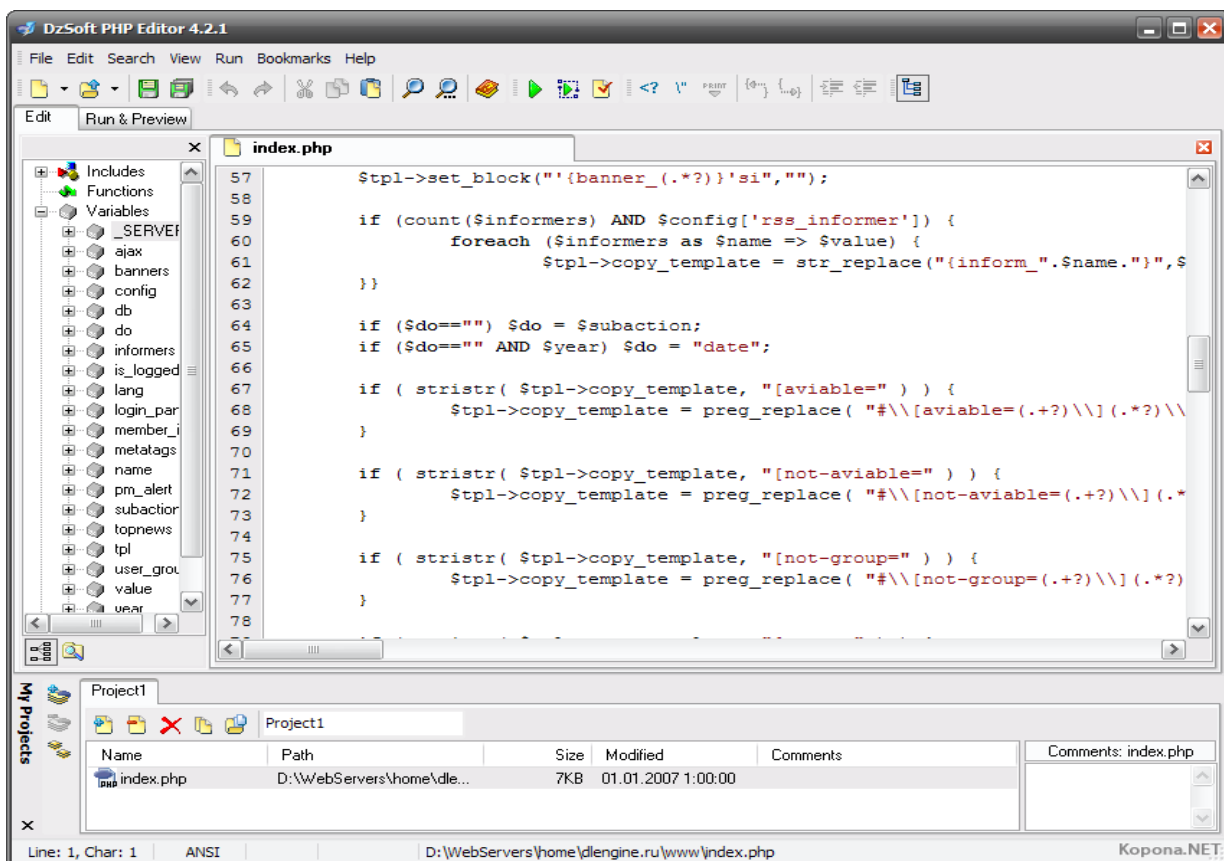


Рисунок 1.6 – Вид головного вікна DzSoft PHP Editor

2 ФОРМУВАННЯ МОДЕЛЕЙ ОБ'ЄКТА ДОСЛІДЖЕННЯ

В першу чергу PHP використовується для створення скриптів, що працюють на стороні сервера, для цього його, власне, і придумали. PHP здатний вирішувати ті ж завдання, що і будь-які інші CGI-скрипти, у тому числі обробляти дані html-форм, динамічно генерувати html сторінки тощо. Але є й інші області, де може використовуватися PHP. Всього виділяють три основні області застосування PHP.

Перша область – це створення додатків (скриптів), які виконуються на стороні сервера. PHP найбільш широко використовується саме для створення такого роду скриптів [19]. Для того щоб працювати таким чином, знадобиться PHP-парсер (тобто обробник php-скриптів) і web-сервер для обробки скрипта, браузер для перегляду результатів роботи скрипта, ну, і, звичайно, який-небудь текстовий редактор для написання самого php-коду. Парсер PHP розповсюджується у вигляді CGI-програми або серверного модуля. Як встановити його і web-сервер на свій комп'ютер, ми розглянемо трохи пізніше. В цьому курсі ми будемо обговорювати, як правило, створення саме серверних додатків, як приклад використання мови PHP.

Друга область – це створення скриптів, що виконуються в командному рядку. Тобто за допомогою PHP можна створювати такі скрипти, які будуть виконуватися, незалежно від web-сервера і браузера, на конкретній машині. Для такої роботи потрібно лише парсер PHP (в цьому випадку його називають інтерпретатором командного рядка (cli, command line interpreter)). Цей спосіб роботи підходить, наприклад, для скриптів, які повинні виконуватися регулярно за допомогою різних планувальників задач або для вирішення задач простої обробки тексту.

І остання область – це створення GUI-додатків (графічних інтерфейсів), що виконуються на стороні клієнта [20]. В принципі це не найкращий спосіб використовувати PHP, особливо для початківців, але якщо ви вже досконально вивчили PHP, то такі можливості мови можуть виявитися вельми корисні.

Для застосування PHP в цій області буде потрібно спеціальний інструмент – PHP-GTK, який є розширенням PHP.

Розглянемо ці області у наступній діаграмі (рис. 2.1).

2.1 Типи даних

PHP підтримує вісім простих типів даних.

Чотири скалярних типи:

- `boolean` (логічний);
- `integer` (цілий);
- `float` (з плаваючою точкою);
- `string` (строковий).

Два змішаних типи:

- `array` (масив);
- `object` (об'єкт).

І два спеціальних типи:

- `resource` (ресурс);
- `NULL`.

В PHP не прийнято явне оголошення типів змінних. Краще, щоб це робив сам інтерпретатор під час виконання програми в залежності від контексту, в якому використовується змінна. Також можна знайти кілька згадувань типу подвійний точності (`double`), в цьому випадку потрібно розглядати його як число з плаваючою точкою, два імені існують тільки з історичних причин.

Якщо потрібно перевірити тип і значення певного виразу, використовують функцію `var_dump()`. Якщо ж для відладки необхідно просто легкому для читання уявлення типу, тоді використовують функцію `gettype()`. Для перевірки на певний тип, не використовуйте `gettype()`, застосовуйте для цього `is_type` функції. Ось кілька прикладів:

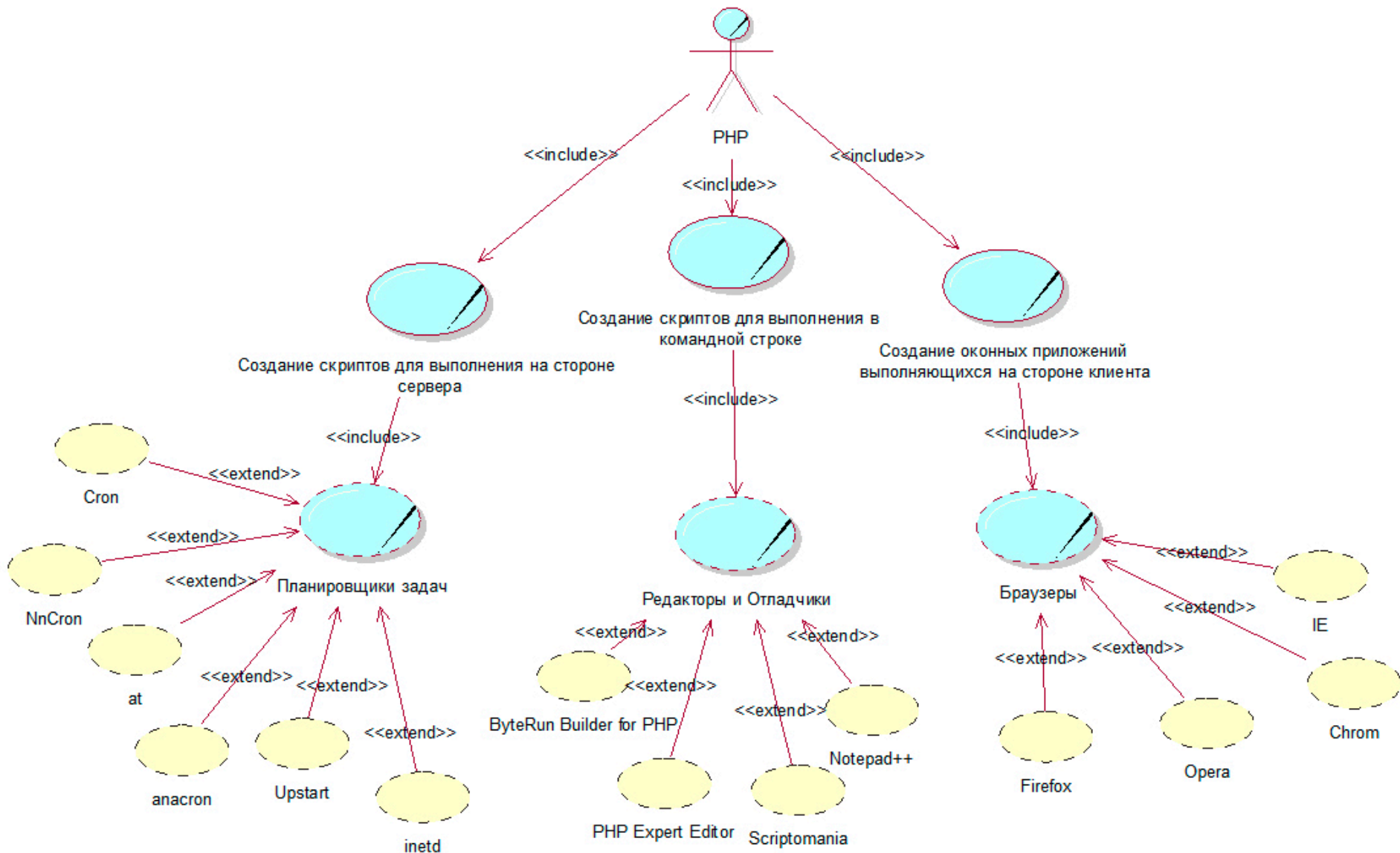


Рисунок 2.1 – Діаграма варіантів використання

```

<? php
$a_bool = TRUE; // логічний
$a_str = "test"; // рядковий
$a_n_int = 12; // цілочисельний
echo gettype ($a_bool); // виводить: boolean
echo gettype ($a_str); // виводить: string
if (is_int ($a_n_int)) { $a_n_int += 4; } // якщо це ціле,
збільшити на чотири
if (is_string ($a_bool)) { echo "Рядок: $a_bool"; } // якщо
$a_bool - це рядок,
// вивести її (Нічого не виводить)
?>

```

Якщо необхідно примусово змінити тип змінної, можна або привести змінну, або використовувати функцію `settype ()`. Зверніть увагу, що змінна, залежно від її типу в даний момент, в певних ситуаціях може мати різні значення.

Розглянемо по порядку всі перераховані типи даних.

1) Тип `boolean` (логічний або логічний тип) – цей найпростіший тип виражає істинність значення, тобто змінна цього типу може мати тільки два значення – істина `TRUE` або брехня `FALSE`. Щоб визначити логічний тип, використовують ключове слово `TRUE` або `FALSE`. Обидва реєстронезалежні.

```

<? php
$test = true; // привласнити $test значення TRUE
?>

```

Логічні змінні використовуються в різних керуючих конструкціях (циклах, умовах тощо). Мати логічний тип, тобто приймати тільки два значення, істину або брехню, можуть також і деякі оператори (наприклад, оператор рівності). Вони також використовуються в керуючих конструкціях для перевірки будь-яких умов. Наприклад, в умовній конструкції перевіряється істинність значення оператора або змінної і залежно від результату перевірки виконуються ті чи інші дії. Тут умова може бути істинно або хибно, що якраз і відображає змінна і оператор логічного типу. Зазвичай, деякий оператор повертає `boolean` значення, яке потім передається керуючій конструкції.

```

<? php
// == це оператор, який перевіряє еквівалентність і
повертає boolean
if ($action == "show_version") { echo "The version is
1.23";}
// це необов'язково ...
if ($show_separators == TRUE) { echo "<hr> \ n";}
// ... тому що наступне має той же самий зміст:
if ($show_separators) { echo "<hr> \ n"; }
?>

```

Для явного перетворення в `boolean`, використовують `(bool)` або `(boolean)`. Однак, в більшості випадків приведення типу необов'язково, тому що значення буде автоматично перетворено, якщо оператор, функція або керуюча конструкція вимагає `boolean` аргумент. При перетворенні в `boolean`, наступні значення розглядаються як `FALSE`:

- саме значення `boolean FALSE`;
- `integer 0` (нуль);
- `float 0.0` (нуль);
- порожній рядок, і рядок `"0"`;
- масив без елементів;
- об'єкт без полів (тільки для PHP 4);
- особливий тип `NULL` (включаючи невстановлені змінні);
- об'єкти `SimpleXML`, створені з порожніх тегів.

Всі інші значення розглядаються як `TRUE` (включаючи будь `resource`). Число `-1` розглядається як `TRUE`, як і будь-яке інше ненульове (негативне або позитивне) число.

2) Тип `integer` (цілі) – цей тип задає число з безлічі цілих чисел $Z = \{..., -2, -1, 0, 1, 2, ...\}$. Цілі можуть бути зазначені в десятковій, шістнадцятковій або вісімковій системі числення, за бажанням з попереднім знаком `"-"` або `"+"`.

Якщо використовується вісімкова система числення, потрібно випередити число `0` (нулем), для використання шістнадцятковій системи потрібно поставити перед числом `0x`. Приклад цілих чисел:


```

<? php
$a = 1234; // десяткове число
$a = -123; // негативне число
$a = 0123; // вісімкове число (еквівалентно 83 у
десятковій системі)
$a = 0x1A; // шістнадцяткове число (еквівалентно 26 у
десятковій системі)
?>

```

Розмір цілого залежить від платформи, хоча, як правило, максимальне значення близько двох мільярдів (це 32-бітне знакове). Беззнакові цілі PHP не підтримує. З версії PHP 4.4.0 і PHP 5.0.5 розмір `integer` може бути визначений за допомогою константи `PHP_INT_SIZE`, а його максимальне значення – за допомогою константи `PHP_INT_MAX`.

Якщо було визначено число, що перевищує межі цілого типу, воно буде інтерпретовано як число з плаваючою крапкою. Також якщо використовується оператор, результатом роботи якого буде число, що перевищує межі цілого, замість нього буде повернуто число з плаваючою крапкою.

В PHP не існує оператора ділення цілих. Результатом $1/2$ буде число з плаваючою точкою 0.5 . В цьому випадку можна привести значення до цілого, що завжди округлює його в меншу сторону, або використовувати функцію `round()`, округлюються значення за стандартними правилами. Для перетворення змінної до конкретного типу потрібно перед змінної вказати в дужках потрібний тип. Наприклад, для перетворення змінної `$a = 0.5` до цілого типу необхідно написати `(integer)(0.5)` або `(integer)$a` або використовувати скорочений запис `(int)(0.5)`. Можливість явного приведення типів за таким принципом існує для всіх типів даних (звичайно, не завжди значення одного типу можна перевести в інший тип).

Якщо число з плаваючою точкою перевищує розміри `integer` (зазвичай $\pm 2.15 \times 10^9 = 2^{31}$ на 32-бітних системах і $\pm 9.22 \times 10^{18} = 2^{63}$ на 64-бітних системах), результат буде невизначеним, так як `float` не має достатньої точності, щоб повернути вірний результат.

В цьому випадку не буде виведено ні попередження, ні навіть зауваження.

3) Тип float (числа з плаваючою точкою) – числа з плаваючою крапкою (вони ж числа подвійної точності або дійсні числа) можуть бути визначені за допомогою будь-якого з наступних синтаксисів:

```
<? php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Розмір числа з плаваючою крапкою залежить від платформи, хоча максимум, як правило, $\sim 1.8e308$ з точністю близько 14 десяткових цифр.

Числа з плаваючою точкою мають обмежену точність. Хоча це залежить від операційної системи, в PHP зазвичай використовується формат подвійної точності IEEE 754, що дає максимальну відносну помилку округлення порядку $1.11e-16$. Неелементарні арифметичні операції можуть давати великі помилки, і, зрозуміло, необхідно брати до уваги поширення помилок при спільному використанні декількох операцій.

Крім того, раціональні числа, які можуть бути точно представлені у вигляді чисел з плаваючою точкою з основою 10, наприклад, 0.1 або 0.7, не мають точного внутрішнього подання в якості чисел з плаваючою точкою з основою 2, незалежно від розміру мантиси. Тому вони й не можуть бути перетворені в їх внутрішню двійкову форму без невеликий втрати точності. Це може привести до несподіваних результатів: наприклад, `floor ((0.1 + 0.7) * 10)` швидше за все поверне 7 замість очікуваного 8, так як результат внутрішнього подання буде чимось на зразок 7.9999999999999991118

Деякі числові операції можуть повертати значення, яке надається константою NAN. Даний результат означає невизначений або невимовне значення в операціях з плаваючою крапкою. Будь суворе або нестроге порівняння даного значення з іншим значенням, включаючи його самого, поверне FALSE.

Так як NAN являє собою необмежену кількість різних значень, то NAN не слід порівнювати з іншими значеннями, включаючи її саму. Замість цього, для визначення її наявності необхідно використовувати функцію `is_nan ()`.

4) Тип `string` (рядки) – це набір символів, де символ – це те ж саме, що й байт. Це означає, що PHP підтримує рівно 256 різних символів, а також те, що в PHP немає вбудованої підтримки Unicode. Немає жодних проблем, якщо рядок дуже велика. Практично не існує обмежень на розмір рядків, що накладаються PHP; єдиний ліміт – це доступна пам'ять на комп'ютері, на якому працює PHP. Рядок може бути визначена чотирма різними способами:

- одинарними лапками;
- подвійними лапками;
- `heredoc`-синтаксисом;
- `nowdoc`-синтаксисом (починаючи з версії PHP 5.3.0).

Одинарні лапки. Найпростіший спосіб визначити рядок - це укласти її в одинарні лапки `' '`. Щоб використовувати одинарні лапки всередині рядка, як і в багатьох інших мовах, перед нею необхідно поставити символ зворотної косою риси `'\'`, тобто екранувати її. Якщо зворотна коса риска повинна йти перед одинарною лапками або бути в кінці рядка, необхідно продублювати її `'\\'`.

Якщо всередині рядка, укладеної в одинарні лапки, зворотний слеш `'\'` зустрічається перед будь-яким іншим символом (відмінним від `'\'` і `'\"'`), то він розглядається як звичайний символ і виводиться, як і всі інші. Тому зворотну косу риску необхідно екранувати, тільки якщо вона знаходиться в кінці рядка, перед закриває лапками.

На відміну від синтаксису подвійних лапок і `heredoc`, змінні і керуючі послідовності для спеціальних символів, укладених в одинарні лапки, не обробляються.

Подвійні лапки. Якщо рядок укладена в подвійні лапки (`"`), PHP розпізнає більшу кількість керуючих послідовностей для спеціальних символів (табл. 2.1).

Таблиця 2.1 – Керуючі послідовності

Послідовність	Значення
1	2
\n	новий рядок (LF або 0x0A (10) в ASCII)
\r	повернення каретки (CR або 0x0D (13) в ASCII)
\t	горизонтальна табуляція (HT або 0x09 (9) в ASCII)
\v	вертикальна табуляція (VT або 0x0B (11) в ASCII) (з версії PHP 5.2.5)
\e	escape-знак (ESC або 0x1B (27) в ASCII) (з версії PHP 5.4.0)
\f	подача сторінки (FF або 0x0C (12) в ASCII) (з версії PHP 5.2.5)
\\	зворотна коса риска
\\$	знак долара
\”	подвійні лапки
\[0-7]{1,3}	послідовність символів, що відповідає регулярному виразу символу в вісімковій системі числення
\x[0-9A-Fa-f]{1,2}	послідовність символів, що відповідає регулярному виразу символу в шістнадцятковій системі числення

Як і в рядку, укладеної в одинарні лапки, екранування будь-якого символу виведе також і саму зворотну косу риску. До версії PHP 5.1.1, зворотна коса риска в `\ {$ var}` не друкувалася. Але найважливішим властивістю рядків в подвійних лапках є обробка змінних.

5) Тип `array`. Масив в PHP – це впорядковане відображення, яке встановлює відповідність між значенням і ключем. Цей тип оптимізований в декількох напрямках, тому ви можете використовувати його як власне масив, список (вектор), хеш-таблицю (що є реалізацією карти), словник, колекцію, стек, чергу і, можливо, щось ще. Так як значенням масиву може бути інший масив PHP, можна також створювати дерева і багатовимірні масиви.

Масив може бути створений мовною конструкцією `array ()`. Як параметри вона приймає будь-яку кількість розділених комами пар `key => value` (ключ => значення).

```
array (key => value, ...)
// key може бути integer або string
// value може бути будь-яким значенням будь-якого типу
```

Мовна конструкція `array ()` приймає як параметри пари ключ => значення, розділені комами. Символ `=>` встановлює відповідність між значенням і його ключем. Ключ може бути як цілим числом, так і рядком, а значення може бути будь-якого наявного в PHP типу. Числовий ключ масиву часто називають індексом. Індування масиву в PHP починається з нуля. Значення елемента масиву можна отримати, вказавши після імені масиву в квадратних дужках ключ шуканого елемента. Якщо ключ масиву являє собою стандартну запис цілого числа, то він розглядається як число, в іншому випадку - як рядок. Тому запис `$a ["1"]` рівносильна запису `$a [1]`, так само як і `$a ["-1"]` рівнозначно `$a [-1]`.

```
<? php
$books = array ("php" =>
                "some book",
                12 => true);
echo $books ["php"];
// виведе "some book"
echo $books [12];
?>
```

Якщо для елемента ключ не заданий, то в якості ключа береться максимальний числовий ключ, збільшений на одиницю. Якщо вказати ключ, якому вже було присвоєно якесь значення, то воно буде перезаписано. Починаючи з PHP 4.3.0, якщо максимальний ключ – негативне число, то наступним ключем масиву буде нуль (0).

```
<? php
// масиви $arr і $arr1 еквівалентні
$arr = array (5 => 43, 32, 56, "b" => 12);
```

```
$arr1 = array (5 => 43, 6 => 32,
              7 => 56, "b" => 12);
?>
```

Якщо використовувати в якості ключа TRUE або FALSE, то його значення переводиться відповідно в одиницю і нуль типу integer. Якщо використовувати NULL, то замість ключа отримаємо порожній рядок. Можна використовувати і саму порожню рядок як ключ, при цьому її треба брати в лапки. Так що це не те ж саме, що використання порожніх квадратних дужок. Не можна використовувати як ключа масиви та об'єкти.

Визначення за допомогою синтаксису квадратних дужок. Створити масив можна, просто записуючи в нього значення. Як ми вже говорили, значення елемента масиву можна отримати за допомогою квадратних дужок, усередині яких потрібно вказати його ключ, наприклад, \$book ["php"]. Якщо вказати новий ключ і нове значення, наприклад, \$book ["new_key"] = "new_value", то в масив додається новий елемент. Якщо ми не вкажемо ключ, а тільки привласнимо значення \$book [] = "new_value", то новий елемент масиву буде мати числовий ключ, на одиницю більший максимального існуючого. Якщо масив, в який ми додаємо значення, ще не існує, то він буде створений.

```
<? php
$books ["key"] = value;
// додали в масив $books значення value з ключем key
$books [] = value1;
// додали в масив значення value1 з ключем 13, оскільки
// максимальний ключ був 12
?>
```

Для того щоб змінити конкретний елемент масиву, потрібно просто присвоїти йому з його ключем нове значення. Змінити ключ елемента не можна, можна тільки видалити елемент (пару ключ / значення) і додати нову. Щоб видалити елемент масиву, потрібно використовувати функцію unset ().

```
<? php
$arr = array (5 => 1, 12 => 2);
```

```

$arr [] = 56; // В цьому місці скрипта це те ж саме, що і
$arr [13] = 56;
$arr ["x"] = 42; // Це додає до масиву новий елемент з
ключем "x"
unset ($arr [5]); // Це видаляє елемент з масиву
unset ($arr); // Це видаляє масив повністю
?>

```

Функція `unset ()` дозволяє видаляти ключі масиву. Зверніть увагу, що масив НЕ буде переіндексований. Якщо ви дійсно хочете поведінки в стилі "видалити і зрушити", можна переіндексувати масив використовуючи `array_values ()`.

```

<? php
$a = array (1 => 'один', 2 => 'два', 3 => 'три');
unset ($a [2]);
/* Дасть масив, представлений так:
   $a = array (1 => 'one', 3 => 'three');
   а НЕ так:
   $a = array (1 => 'one', 2 => 'three'); */
$b = array_values ($a); // Тепер $b це array (0 => 'one',
1 => 'three')
?>

```

Для кожного з типів: `integer`, `float`, `string`, `boolean` та `resource`, перетворення значення в масив дає результатом масив з одним елементом (з індексом 0), що є скалярним значенням, з якого ви почали. Іншими словами, `(array) $ scalarValue` – це точно те ж саме, що і `array ($ scalarValue)`.

Якщо ви перетворюєте в масив об'єкт (`object`), ви отримаєте як елементи масиву властивості (змінні-члени) цього об'єкта. Ключами будуть імена змінних-членів, з деякими примітними виключеннями: цілочисельні властивості стануть недоступні; до закритих полів класу (`private`) спереду буде дописано ім'я класу; до захищених полям класу (`protected`) спереду буде додано символ `'*`'. Ці додані значення з обох сторін також мають нульові байти. Це може викликати дещо несподіване поведінку.

б) Тип `object`. Об'єкти – тип даних, що прийшов з об'єктно-орієнтованого програмування (ООП). Згідно з принципами ООП, клас – це набір об'єктів, що володіють певними властивостями і методами роботи з ним, а об'єкт відповідно – екземпляр класу. Наприклад, програмісти – це клас

людей, які пишуть програми, вивчають комп'ютерну літературу і, крім того, як усі люди, мають ім'я та прізвище. Тепер, якщо взяти одного конкретного програміста, Васю Іванова, то можна сказати, що він є об'єктом класу програмістів, має ті ж властивості, що й інші програмісти, теж має ім'я, пише програми і т.п.

У PHP для доступу до методів об'єкта використовується оператор `->`. Для ініціалізації об'єкта використовується вираз `new`, що створює в змінній екземпляр об'єкта.

```
<? php
class Person // створюємо клас людей
{
// Метод, який навчає людину робить якусь дію
function action () { echo "це функція"; }
}
$bob = new Person; // створюємо об'єкт класу людина
$bob -> action (); // робимо дію
?>
```

Якщо `object` перетвориться в `object`, він не змінюється. Якщо значення іншого типу перетвориться в `object`, створюється новий екземпляр вбудованого класу `stdClass`. Якщо значення було `NULL`, новий екземпляр буде порожнім. Масиви перетворюються в `object` з іменами полів, названими згідно ключам масиву і відповідними їм значеннями. При перетворенні будь-якого іншого значення, воно буде поміщено в поле з ім'ям `scalar`.

```
<? php
$ obj = (object) 'ciao';
echo $ obj-> scalar; // виведе 'ciao'
?>
```

7) Тип `resource`. Ресурс – це спеціальна змінна, що містить посилання на зовнішній ресурс (наприклад, з'єднання з базою даних). Ресурси створюються і використовуються спеціальними функціями (наприклад, `mysql_connect ()`, `pdf_new ()` тощо).

Функція `get_resource_type` повертає тип даного ресурсу. Якщо цей параметр `handle` є ресурсом, функція повертає рядок, що вказує його тип. Якщо

тип не визначається цією функцією, повертається значенням буде рядок Unknown. Функція повертає FALSE і викликає помилку, якщо handle не є ресурсом (resource). Приклад використання get_resource_type ():

```
<? php
// Виводить: mysql link
$c = mysql_connect ();
echo get_resource_type ($ c). "\ n";
// Виводить: file
$ fp = fopen ("foo", "w");
echo get_resource_type ($ fp). "\ n";
// Виводить: domxml document
$ doc = new_xml doc ("1.0");
echo get_resource_type ($ doc-> doc). "\ n";
?>
```

Оскільки тип resource містить спеціальні вказівники на відкриті файли, з'єднання з базою даних, області зображення тощо, перетворення в цей тип не має сенсу. Завдяки системі підрахунку посилань, введеної в PHP 4 Zend Engine, визначення відсутності посилань на ресурс відбувається автоматично, після чого він звільняється складальником сміття. Тому, дуже рідко потрібно звільняти пам'ять вручну. Постійні з'єднання з базами даних є винятком із цього правила. Вони не знищуються збирачем сміття.

8) Тип Null. Спеціальне значення NULL говорить про те, що змінна не має значення. Мінлива вважається NULL, якщо:

- їй була присвоєна константа NULL (\$ var = NULL);
- їй ще не було присвоєно яке-небудь значення;
- вона була видалена за допомогою unset ().

Існує тільки одне значення типу NULL – регістронезалежне ключове слово NULL.

```
<? php
$ var = NULL;
?>
```

Для перевірки, чи є значення змінної рівним NULL використовується функція `is_null`.

```
bool is_null (mixed $ var)
```

Вона повертає TRUE, якщо значення `var` одно `null`, або FALSE в протилежному випадку. Приклад використання `is_null ()`:

```
<? php
error_reporting (E_ALL);

$test = NULL;
var_dump (is_null ($inexistent), is_null ($test));
?>
```

Внаслідок можна побачити наступне:

```
Notice: Undefined variable: inexistent in ...
bool (true)
bool (true)
```

2.2 Оператори в середовищі розробки PHP

Оператори та функції є основними інструментами будь-якої мови програмування, в тому числі і PHP. З їх допомогою можна здійснювати управління послідовністю дій в ході виконання сценарію. До основних функцій мови PHP можна віднести математичні функції і функції роботи з даними.

До основних операторів мови PHP відносяться оператори обробки даних, такі як математичні, оператори присвоювання, оператори порівняння, логічні оператори. Крім цього, у мові PHP використовуються керуючі конструкції мови – умовні оператори і оператори циклів.

До основних математичних операторів PHP відносяться наступні.

Операція ділення ("/") завжди повертає дійсний тип, навіть якщо обидва значення були цілочисельними (або рядками, які перетворюються в цілі числа). В іншому випадку результат буде дрібним.

Таблиця 2.2 – Математичні оператори

Приклад	Назва	Результат
$-\$a$	Заперечення	Зміна знака $\$a$
$\$a + \b	Додавання	Сума $\$a$ і $\$b$
$\$a - \b	Віднімання	Різниця $\$a$ і $\$b$
$\$a * \b	Множення	Добуток $\$a$ і $\$b$
$\$a / \b	Ділення	Частка від ділення $\$a$ на $\$b$
$\$a \% \b	Ділення по модулю	Цілочисельний залишок від ділення $\$a$ на $\$b$.

Операція обчислення залишку від ділення "%" працює тільки з цілими числами, так що застосування її до дробовим може привести до небажаного результату. Залишок $\$a \% \b буде негативним, для негативних значень $\$a$.

Можливе використання дужок. Пріоритет одних математичних операцій над іншими і зміна пріоритетів при використанні дужок в арифметичних виразах відповідають звичайним математичним правилам.

Базовий оператор присвоєння позначається як $=$. На перший погляд може здатися, що це оператор "дорівнює". Насправді це не так. В дійсності, оператор присвоєння означає, що лівий операнд отримує значення правого вираження, (тобто встановлюється результуючим значенням).

Результатом виконання оператора присвоєння є саме присвоєне значення. Таким чином, результат виконання $\$a = 3$ буде дорівнювати 3. Це дозволяє використовувати конструкції виду:

```
<?php
$a = ($b = 4) + 5;
// результат: $a встановлена значенням 9, змінної $b
присвоєно 4.
?>
```

Оператори порівняння, як це видно з їхньої назви, дозволяють порівнювати між собою два значення. Це у своєму роді унікальні операції, тому

що незалежно від типів своїх аргументів вони завжди повертають одне з двох: false або true. Операції порівняння дозволяють порівнювати два значення між собою і, якщо умова виконана, повертають true, а якщо ні – false.

В PHP дозволяється порівнювати тільки скалярні змінні. Масиви та об'єкти в PHP порівнювати не можна. Їх навіть не можна порівнювати на рівність (за допомогою оператора ==), але при виконанні такої операції PHP не видає попередження. Так що здивувавшись якось раз, чому два абсолютно різних масиви при порівнянні їх з допомогою == виявляються раптом однаковими, згадайте, що перед порівнянням обидва операнда перетворюються в слово array, яке потім і порівнюється. Розглянемо оператори порівняння у табл. 2.3.

Таблиця 2.3 – Таблиця операторів порівняння

Приклад	Назва	Результат
<code>\$a == \$b</code>	Дорівнює	TRUE якщо \$a дорівнює \$b.
<code>\$a === \$b</code>	Тотожно дорівнює	TRUE якщо \$a дорівнює \$b і має той же тип.
<code>\$a != \$b</code>	Не дорівнює	TRUE якщо \$a не дорівнює \$b.
<code>\$a <> \$b</code>	Не дорівнює	TRUE якщо \$a не дорівнює \$b.
<code>\$a !== \$b</code>	Тотожно не дорівнює	Тотожно не дорівнює TRUE якщо \$a не дорівнює \$b або у випадку, якщо вони різних типів
<code>\$a < \$b</code>	Менше	TRUE якщо \$a строго менше \$b.
<code>\$a > \$b</code>	Більше	TRUE якщо \$a строго більше \$b.
<code>\$a <= \$b</code>	Менше або дорівнює	TRUE якщо \$a менше або дорівнює \$b.
<code>\$a >= \$b</code>	Більше або дорівнює	TRUE якщо \$a більше або дорівнює \$b.

У випадку, якщо ви порівнюєте ціле з рядком, рядок буде перетворена до числа. У випадку, якщо ви порівнюєте дві числові рядки, вони порівнюються як цілі числа.

Логічні оператори призначені виключно для роботи з логічними виразами і повертають false або true. Наведемо таблицю логічних операторів PHP (табл. 2.4).

Сенс двох різних варіантів для операторів "and" і "or" в тому, що вони працюють з різними пріоритетами.

Слід зауважити, що обчислення логічних виразів, що містять такі оператори, йде завжди зліва направо, при цьому, якщо результат уже очевидний (наприклад, false && щось завжди дає false), то обчислення обриваються, навіть якщо у виразі присутні виклики функцій.

Таблиця 2.4 – Логічні оператори

Приклад	Назва	Результат
<code>\$a and \$b</code>	Логічне «і»	TRUE якщо і <code>\$a</code> , і <code>\$b</code> TRUE
<code>\$a or \$b</code>	Логічне «або»	TRUE якщо або <code>\$a</code> , або <code>\$b</code> TRUE
<code>\$a xor \$b</code>	Виключне «або»	TRUE якщо <code>\$a</code> , або <code>\$b</code> TRUE, але не обидва
<code>! \$a</code>	Заперечення	TRUE якщо <code>\$a</code> НЕ TRUE
<code>\$a && \$b</code>	Логічне «і»	TRUE якщо і <code>\$a</code> , і <code>\$b</code> TRUE
<code>\$a \$b</code>	Логічне «або»	TRUE якщо або <code>\$a</code> , або <code>\$b</code> TRUE

У випадку з логічними операціями не варто забувати про подвоєння символу. Наприклад, `| та ||` – два зовсім різних оператора, один з яких може потенційно повертати будь-яке число, а другий – тільки false і true.

3 РЕЗУЛЬТАТИ ТЕОРЕТИЧНИХ ТА ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

3.1 Налаштування конфігурації

Для доступу до важливих змінних зазвичай створюють окремі файли з константами і з настройками конфігурації. Це потрібно для того, щоб змінити якісь змінні при необхідності. Припустимо потрібно щоб в таблиці рейтингу відвіданих сторінок було не 10 елементів як ми вказали в скрипті, а 20. Без файлу конфігурації довелося б шукати ту змінну яка відповідає за цю величину в одній з функцій, і на це зник б якийсь час. Маючи файл з такого роду змінними ми значно спрощуємо роботу з цим сценарним додатком. Файл конфігурації можна знайти за адресою `/conf/config.php`.

Параметри у файлі конфігурації можна поділити на 3 підгрупи:

- параметри, що впливають на візуальне подання;
- параметри, що впливають на функціональність;
- різні параметри.

Нижче розглянемо параметри і синтаксис цих параметрів для кожної підгрупи:

1) Параметри, що впливають на візуальне подання.

- `$CNT_MAINSITE = "..";`

Якщо ця змінна встановлена, посилання на вказане місце буде згенеровано. Типово значення вказує на батьківську директорію. У випадку, якщо сайт розташований в іншому місці, необхідно змінити значення відповідно з адресою.

Синтаксис:

Посилання на вказану адресу

```
$CNT_MAINSITE = "http://www.myserver.com/";
```

Посилання на батьківську директорію (типово)

```
$CNT_MAINSITE = ".."
```

Не показувати посилання

```
$CNT_MAINSITE = "";
```

```
- $CNT_SHOW_CONFIG = "";
```

Додаток CNT типово показує настройки статистики. У випадку, якщо у відображенні налаштувань немає потреби, можна відключити цю опцію.

Синтаксис:

Відображати show_config.php

```
$CNT_SHOW_CONFIG = 1;
```

Не відображати show_config.php (типово)

```
$CNT_SHOW_CONFIG = "";
```

```
- $CNT_TITLEBAR = "";
```

Змінна заголовка, який буде відображатись в навігаційній панелі всіх веб-сторінок статистики. Можна використовувати наступні макроси для відображення на сторінках: %SERVER - ім'я сервера, %DATE - поточна дата.

Синтаксис:

"Статистика для %SERVER згенерована %DATE" (типово)

```
$CNT_TITLEBAR = "";
```

Фіксований текст, не для конкретної мови

```
$CNT_TITLEBAR = "Какой то текст";
```

```
- $CNT_LANGUAGE = "ru";
```

Змінна мови що використовується за умовчанням, в разі, якщо вона не була вказана в браузері. Для коректної роботи мова повинна бути встановлена обов'язково. Підтримуються наступні мови: en, ru.

– `$CNT_MAXTIME = 1800;`

Ця змінна визначає тривалість унікального відвідування у секундах. Кожен хіт з того ж відвідувача протягом цього періоду буде вважатися за одне відвідування до тих пір, поки між двома хітами не буде перевищена зазначена межа. Типово 30 хвилин (1800 секунд). Можна призначити будь-яке інше значення.

Синтаксис:

Кожен хіт унікальний

`$CNT_MAXTIME = 0;`

Кожен хіт протягом 30 хвилин від того ж відвідувача як унікальне відвідування

`$CNT_MAXTIME = 1800;`

– `$CNT_MAXVISIBLE = 100;`

Ця змінна визначає яку кількість значень потрібно відобразити в детальной статистиці. Типово значення дорівнює 100. Не рекомендується виставляти це значення більше 500 щоб уникнути зайвого навантаження. Значення "0" скинути докладну статистику.

Наступні записи вказують скільки пунктів буде відображатися в кожному рейтингу загальної статистики `show_global.php`. У першій групі йдуть змінні максимальної кількості браузерів, країн (розширень), операційних систем і роботів. У другій максимальна кількість хостів, ключових слів, реферерів і сторінок. При непотрібності другу групу можна обнулити встановивши "0" в якості значення.

– `$CNT_MAXBROWSER = 10;`

– `$CNT_MAXEXTENSION = 20;`

– `$CNT_MAXOS = 10;`

– `$CNT_MAXROBOT = 10;`

– `$CNT_MAXHOST = 10;`

– `$CNT_MAXKEY = 10;`

- `$CNT_MAXORIGIN = 10;`
- `$CNT_MAXPAGE = 10;`
- `$CNT_DETAILED_STAT_FIELDS = "id, time, ext, dns, ip, visits, os, browser, referer";`

Ця змінна містить в собі список всіх колонок які повинні відображатися в детальній статистиці. Розглянемо всі доступні варіанти колонок в табл. 3.1.

Таблиця 3.1 – Варіанти доступних колонок для детальної статистики

Назва	Опис
id	N-ий відвідувач з моменту відліку
time	Час коли останній хіт був зареєстрований
visits	Хіти унікальних відвідувань
dns	Ім'я хоста відвідувача
ip	Ip адреса відвідувача
os	Операційна система (якщо доступна і / або не робот)
browser	Програмне забезпечення, що використовується для встановлення з'єднання
ext	Країна або розширення відвідувача
referer	Посилання, з якою прийшов відвідувач (якщо доступне)
page	Остання відвідана сторінка
search	Пошуковий запит відвідувача (якщо доступний)

Порядок пунктів в списку також використовується для організації стовпця у детальній статистиці.

Синтаксис:

```
$CNT_DETAILED_STAT_FIELDS = "id, time, visits, ip, ext, os, browser";
```

```
$CNT_DETAILED_STAT_FIELDS = "date, ext, browser, os, ip";
```

2) Параметри, що впливають на функціональність.

- `$CNT_USE_LOCK = "flk";`

Перед записом у файл, буде спроба блокування файлу, щоб запобігти його від пошкоджень. Якщо скрипт не зможе заблокувати файли, можна встановити альтернативний метод за допомогою опції цієї змінної. Для правильної роботи слід використовувати один з методів блокування. Можливі опції "dio", "flk" і "sem"

Синтаксис:

Використовувати flock() (типово)

```
$CNT_USE_LOCK = "flk";
```

Використовувати SysV семафори (PHP >= 4.1.0 й тільки Unix)

```
$CNT_USE_LOCK = "sem";
```

Використання прямого введення / виводу (PHP >= 4.2.0 й тільки Unix)

```
$CNT_USE_LOCK = "dio";
```

– \$CNT_TIME_OFFSET = 0;

Якщо час сервера не збігається з Вашим локальним часом, ви можете налаштувати час в хвилинах за допомогою цієї змінної. Негативні значення переведуть час назад, позитивні зрушать вперед.

Синтаксис:

Встановити час вперед на 5 годин (5 x 60 min)

```
$CNT_TIME_OFFSET = 300;
```

Встановити час назад на 5 годин (5 x 60 min)

```
$CNT_TIME_OFFSET = -300;
```

Використовувати час сервера "як є" (типово)

```
$CNT_TIME_OFFSET = 0;
```

– \$CNT_NO_DNS = "";

Ця змінна визначає, чи повинні бути дозволені імена хостів в ір адресі. У той час як імена хостів говорять про відвідувача гараздо більше, їх використання може значно сповільнити роботу вашого сайту, якщо використовувати

DNS-сервери повільно працюють, обмежені у можливостях або ненадійні з інших причин. Встановлення цієї опції може вирішити проблему.

Синтаксис:

Дозвіл відображення адреси в імені хоста (типово)

```
$CNT_NO_DNS = "";
```

Заборонити відображення адреси

```
$CNT_NO_DNS = 1;
```

```
– $CNT_EXT_LOOKUP = "";
```

Ця змінна визначає, чи потрібно використовувати IP-адреси, щоб визначити місцезнаходження відвідувача (країну).

Синтаксис:

Ніяких зовнішніх переглядів ір адреси (типово)

```
$CNT_EXT_LOOKUP = "";
```

Використовувати старий IP2EXT для перегляду ір адреси

```
$CNT_EXT_LOOKUP = "IP2EXT";
```

Використовувати GeoIP PHP API плагін

```
$CNT_EXT_LOOKUP = "GEOIP";
```

```
– $CNT_GEOIP_PATH = "cnt/geoip/";
```

При використанні в попередній змінній опції «GEOIP» необхідно вказати шлях до плагіну GeoIP PHP API.

```
– $CNT_NO_HITS = "";
```

Типово показуються хіти у статистиці за часом, тому що це дає корисне враження про фактичне завантаження сервера. Однак, якщо потрібно врахувати тільки унікальних відвідувачів в якості базової статистики за часом, встановивши цю змінну можна змінити засіб підрахунку.

Синтаксис:

Показувати хіти у статистиці за часом (типово)

```
$CNT_NO_HITS = "";
```

Показувати унікальних відвідувачів в статистиці за часом

```
$CNT_NO_HITS = 1;
```

```
– $CNT_IGNORE_IP = "";
```

Опція цієї змінної може використовуватися для виключення деяких ір адрес або діапазонів адрес. У випадку, якщо необхідно додати декілька виразів, потрібно використовувати кому як роздільник. Ця опція не приймає ніяких імен хоста, тільки ір адреси. Можна, однак, опустити частини справа наліво, щоб заблокувати ір діапазон (як показано в прикладі нижче).

Синтаксис:

Ігнорування заданого діапазону

```
$CNT_IGNORE_IP = "127., 192.168.";
```

Нічого не ігнорувати (типово)

```
$CNT_IGNORE_IP = "";
```

```
– $CNT_IGNORE_REFERER = "";
```

У випадку, якщо ви не хочете відобразити в рейтингу або в детальній статистиці реферерів з конкретних сайтів, можна задати список ігнорованих. Кожен реферер буде звірятися з цим списком. Використовується два визначення режимів:

а) якщо вираз починається з косою риси, то воно буде розглядатися як частина адреси реферера. Це корисно для блокування небажаних сайтів безкоштовних хостингів.

б) якщо вираз не починається з косою риси, то воно буде порівнюватися тільки з хостом реферера. Це корисно для визначення небажаних хостів, без випадкового блокування законних реферерів відвідувачів.

Синтаксис:

Ігнорування зазначених збігів

```
$CNT_IGNORE_REFERER = "spambot.org, .escort.";
```

Нічого не ігнорувати (типово)

```
$CNT_IGNORE_REFERER = "";
```

```
– $CNT_HITS = 1;
```

Змінна яка дає змогу включити хіти в статистику.

Синтаксис:

Включити хіти (типово)

```
$CNT_HITS = 1;
```

Ігнорувати хіти

```
$CNT_HITS = "";
```

3) Різні параметри

```
– $CNT_CUSTOM_CHARSET = "";
```

За допомогою цієї опції можна змінити поточне кодування на бажане.

Наприклад:

Використовувати кодування UTF-8

```
$CNT_CUSTOM_CHARSET = "UTF-8";
```

Використовувати стандартне кодування

```
$CNT_CUSTOM_CHARSET = "";
```

```
– $CNT_CSS_FILE = "default.css";
```

Змінна яка містить ім'я CSS файлу стилів.

```
– $CNT_LOADTIME = "1";
```

Ця змінна дозволяє відображати час швидкості завантаження сторінки (в секундах).

Синтаксис:

Не генерувати (типово)

```
$CNT_LOADTIME = "";
```

Генерувати швидкість завантаження внизу сторінки

```
$CNT_LOADTIME = 1;
```

– \$CNT_IGNORE_BOTS = 1;

Щоб визначити обробку роботів можна використовувати цю змінну. Типово вони ігноруються в рейтингу хостів, але залишаються в статистиці. Якщо ви не хочете бачити роботів зовсім, Ви можете встановити опцію цієї змінної на "2", тоді тільки візити людини будуть прийняті до уваги.

Синтаксис:

Повністю ігнорувати роботів

```
$CNT_IGNORE_BOTS = 2;
```

Не відображати імена хостів роботів (типово)

```
$CNT_IGNORE_BOTS = 1;
```

Враховувати роботів як звичайних відвідувачів

```
$CNT_IGNORE_BOTS = "";
```

– \$CNT_IGNORE_AGENT = "";

Ця змінна визначає, як сценарний додаток відрізняє одного відвідувача від іншого. Типово використовується тільки ір адреса, для створення реалістичної картини. Однак, якщо відвідувачі часто захищені за проксі-серверами, деактивація цієї опції забезпечить кращу картину, так як новий відвідувач буде враховуватися, в той час як агент користувача не змінився.

Синтаксис:

Ігнорувати різних агентів одного відвідувача

```
$CNT_IGNORE_AGENT = 1;
```

Враховувати кожного агента користувача (типово)

```
$CNT_IGNORE_AGENT = "";
```

3.2 Результати досліджень

Розроблений сценарний додаток відрізняється від своїх аналогів відсутністю великих баз даних. Всі дані зберігаються в окрему папку для можливості подальшого вилучення цих даних звідти. Завдяки цьому значно скорочується час обробки запитів щодо виведення даних в таблиці і виключаються деякі помилки доступу виникають при роботі з зовнішніми БД. При великому навантаженні на сайт, щоб позбутися від помилок записи можна змінювати кількість файлів призначених для запису даних. При фіксуванні користувача і запису його даних у файл, потрібно цей файл тимчасово заблокувати щоб уникнути можливих помилок запису.

Щоб простежити всі етапи фіксування користувача, починаючи від перевірки адреси і переведення даних реферера в масив, до виведення всіх отриманих даних в випадково обраний файл, створимо діаграму активності дозволяє переглянути стан на різних етапах трасування файлу `marker.php`.

На рис. 3.1 видно як при появі відвідувача спрацьовують алгоритми, що дозволяють визначити інформацію про цього користувача. Як тільки програма зафіксувала користувача, проходять перевірки його ір-адреси, хоста, визначення реферального посилання в проксі заголовку та інші перевірки. Після всіх перевірок вся отримана інформація записується до одного з раніше створених файлів даних.

Для демонстрації досягнутих результатів, розглянемо сторінки статистики які дають змогу спостерігати за зміною активності користувачів на веб-сайті.

На рис. 3.2 зображена веб-сторінка загальної статистики (`show_global.php`), на якому можна побачити рейтинги браузерів, операційних систем, країн та розширень, робіт, хостів, відвіданих сторінок, джерел та ключових слів. Під рейтингами можна побачити загальну статистику відвідувань користувачів.

На рис. 3.3 можна побачити таблицю, що містить повну інформацію про кожного відвідувача веб-сайта, яка формує детальну статистику (`show_detailed.php`). За допомогою змінної `$CNT_DETAILED_STAT_FIELDS` в настройках конфігурації можна вивести необхідну нам інформацію і зайву опустити.

На рис. 3.4 показана сторінка зі статистикою за часом (`show_time.php`). Використовуючи інформацію з загальної статистики можна побудувати кілька графіків, які зможуть показати кількісну різницю відвідувань користувачів. По першому графіку можна подивитися в який час користувачі були активніше. У другому графіку видно в який день тижня користувачі частіше відвідують сайт. У третьому графіку показано в який день місяця користувачі відвідували веб-сайт частіше, а в четвертому в якому місяці.

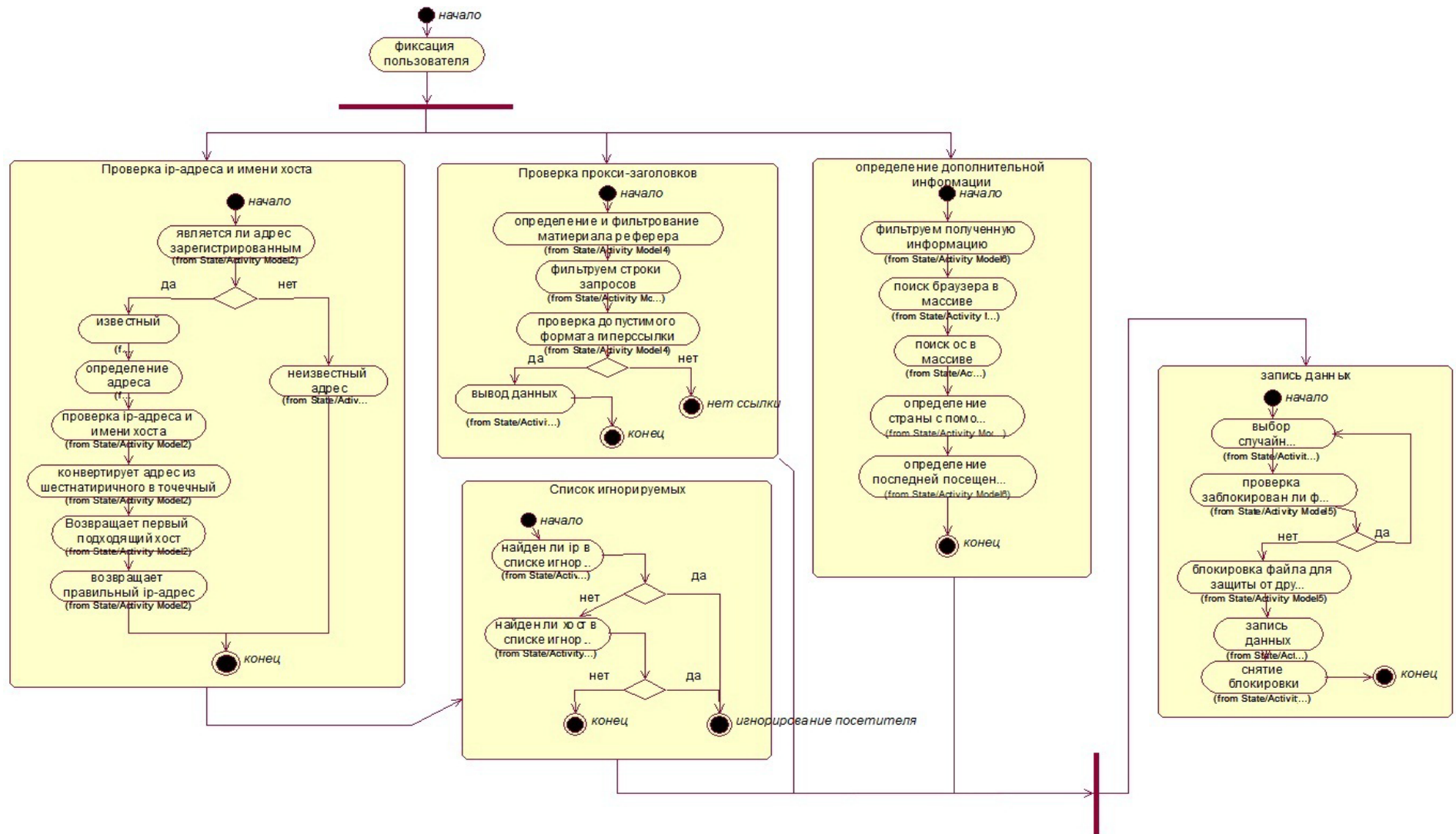


Рисунок 3.1 – Диаграмма активности

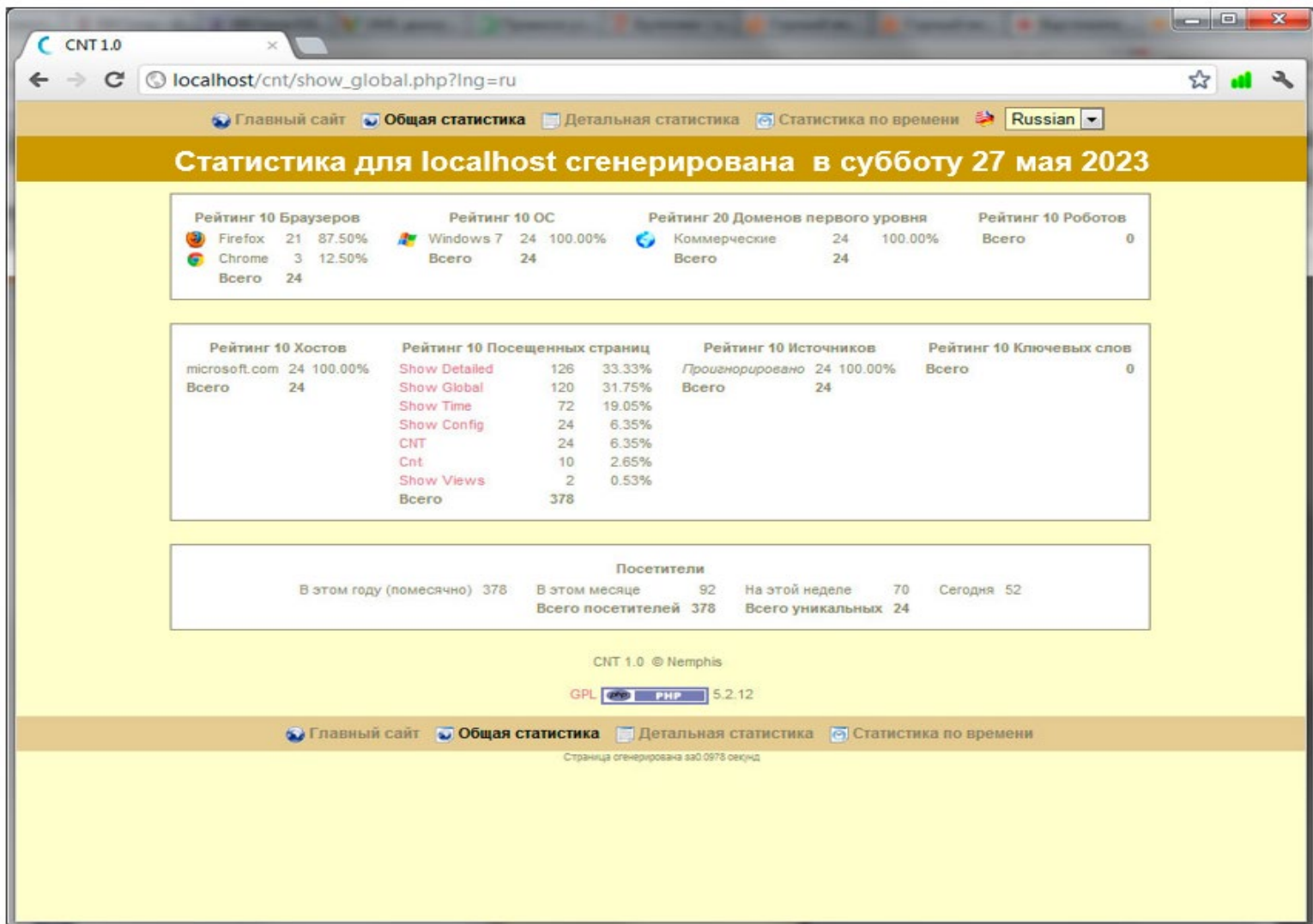


Рисунок 3.2 – Основная статистика

CNT 1.0

localhost/cnt/show_detailed.php?lng=ru

Главный сайт | Общая статистика | **Детальная статистика** | Статистика по времени | Russian

Статистика для localhost сгенерирована в субботу 27 мая 2023

Последних посетителей: 100 | последнее посещение < 1800 сек. | последнее посещение > 1800 сек. | роботы | Посещения с вашего IP

ID	Время	Страна	Адрес посетителя	IP Адрес	Посещения	ОС	Браузер	По ссылке
24	5 Май, 21:00:30	Коммерческие	validation.sls.microsoft.com	127.0.0.1	39	Windows 7	Chrome 18.0.1025.	
23	5 Май, 20:37:20	Коммерческие	validation.sls.microsoft.com	127.0.0.1	14	Windows 7	Firefox 12.0	
22	30 Апр, 19:59:41	Коммерческие	validation.sls.microsoft.com	127.0.0.1	3	Windows 7	Firefox 11.0	
21	30 Апр, 17:15:19	Коммерческие	validation.sls.microsoft.com	127.0.0.1	15	Windows 7	Firefox 11.0	
20	27 Апр, 19:52:37	Коммерческие	validation.sls.microsoft.com	127.0.0.1	1	Windows 7	Firefox 11.0	
19	27 Апр, 14:21:37	Коммерческие	validation.sls.microsoft.com	127.0.0.1	2	Windows 7	Firefox 11.0	
18	20 Апр, 22:27:12	Коммерческие	validation.sls.microsoft.com	127.0.0.1	13	Windows 7	Chrome 18.0.1025.	
17	20 Апр, 18:13:07	Коммерческие	validation.sls.microsoft.com	127.0.0.1	6	Windows 7	Chrome 18.0.1025.	
16	16 Мар, 19:01:09	Коммерческие	validation.sls.microsoft.com	127.0.0.1	5	Windows 7	Firefox 10.0.2	
15	16 Мар, 17:57:57	Коммерческие	validation.sls.microsoft.com	127.0.0.1	3	Windows 7	Firefox 10.0.2	
14	16 Мар, 16:15:21	Коммерческие	validation.sls.microsoft.com	127.0.0.1	3	Windows 7	Firefox 10.0.2	
13	16 Мар, 15:40:41	Коммерческие	validation.sls.microsoft.com	127.0.0.1	1	Windows 7	Firefox 10.0.2	
12	14 Мар, 21:59:02	Коммерческие	validation.sls.microsoft.com	127.0.0.1	2	Windows 7	Firefox 10.0.2	
11	14 Мар, 19:56:20	Коммерческие	validation.sls.microsoft.com	127.0.0.1	35	Windows 7	Firefox 10.0.2	
10	14 Мар, 18:30:00	Коммерческие	validation.sls.microsoft.com	127.0.0.1	12	Windows 7	Firefox 10.0.2	
9	14 Мар, 17:43:16	Коммерческие	validation.sls.microsoft.com	127.0.0.1	2	Windows 7	Firefox 10.0.2	
8	14 Мар, 16:36:08	Коммерческие	validation.sls.microsoft.com	127.0.0.1	1	Windows 7	Firefox 10.0.2	
7	14 Мар, 15:59:24	Коммерческие	validation.sls.microsoft.com	127.0.0.1	12	Windows 7	Firefox 10.0.2	
6	14 Мар, 13:58:12	Коммерческие	validation.sls.microsoft.com	127.0.0.1	64	Windows 7	Firefox 10.0.2	
5	14 Мар, 01:29:48	Коммерческие	validation.sls.microsoft.com	127.0.0.1	98	Windows 7	Firefox 10.0.2	
4	14 Мар, 00:26:17	Коммерческие	validation.sls.microsoft.com	127.0.0.1	30	Windows 7	Firefox 10.0.2	
3	13 Мар, 22:34:27	Коммерческие	validation.sls.microsoft.com	127.0.0.1	4	Windows 7	Firefox 10.0.2	
2	13 Мар, 21:57:37	Коммерческие	validation.sls.microsoft.com	127.0.0.1	7	Windows 7	Firefox 10.0.2	
1	13 Мар, 20:17:13	Коммерческие	validation.sls.microsoft.com	127.0.0.1	7	Windows 7	Firefox 10.0.2	

CNT 1.0 © Nemphis

GPL РНР 5.2.12

Главный сайт | Общая статистика | **Детальная статистика** | Статистика по времени

Страница сгенерирована за 0.22035 секунд

Рисунок 3.3 – Детальная статистика

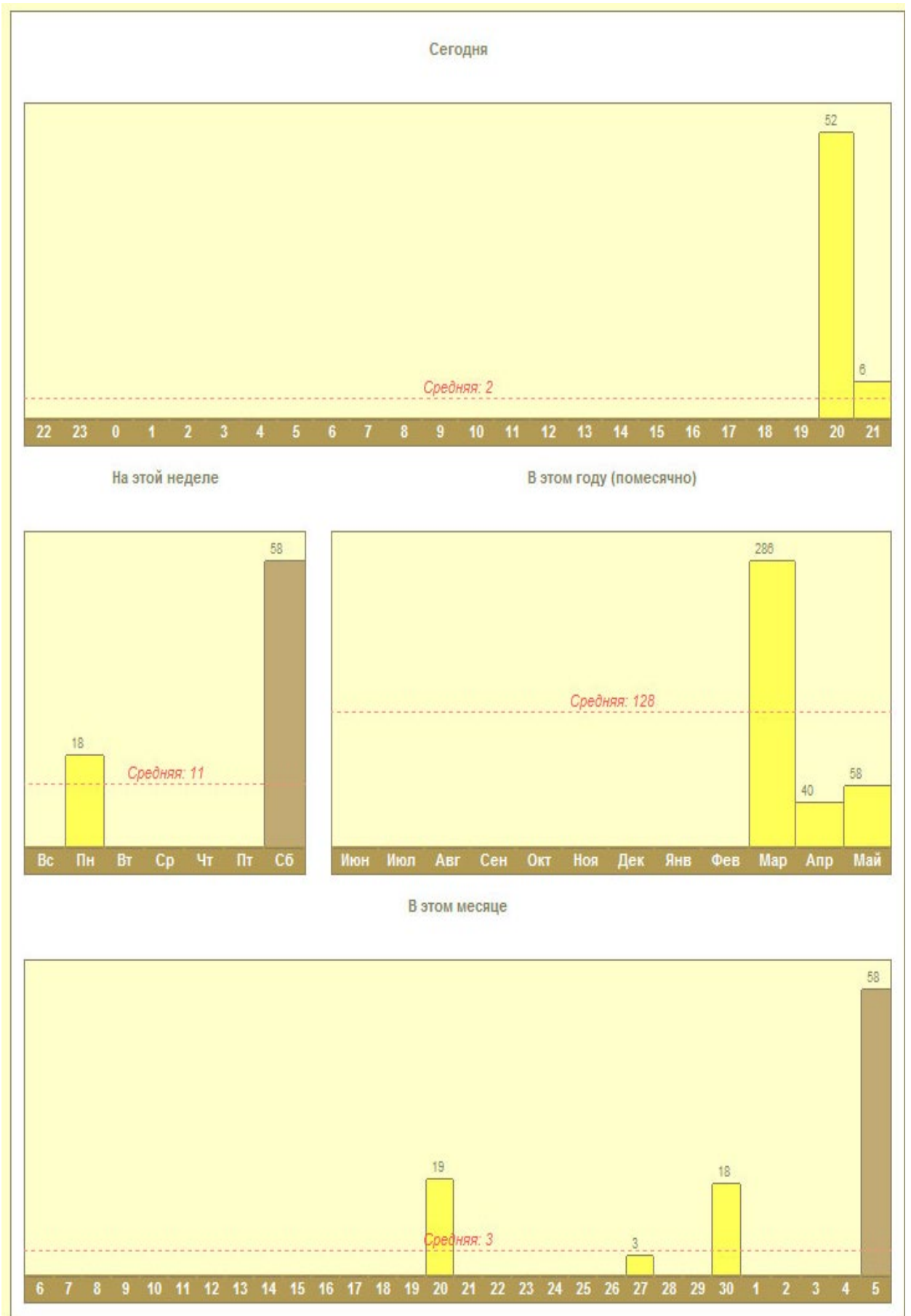


Рисунок 3.4 – Статистика за часом

ВИСНОВКИ

PHP здатний генерувати не тільки HTML. Доступно формування зображень, файлів PDF і навіть роликів Flash (з використанням libswf і Ming), створюваних «на льоту». PHP також здатний генерувати будь-які текстові дані, такі, як XHTML та інші XML-файли. PHP може здійснювати автоматичну генерацію таких файлів і зберігати їх у файлової системі вашого сервера замість того, щоб віддавати клієнту, організовуючи, таким чином, серверний кеш для вашого динамічного контенту.

PHP також підтримує «спілкування» з іншими сервісами с через такі протоколи, як LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) і багатьох інших. Крім того, ви отримуєте можливість працювати з мережевими сокетом безпосередньо. PHP підтримує стандарт обміну складними структурами даних WDDX практично між всіма мовами веб-програмування. Звертаючи увагу на взаємодію між різними мовами, слід згадати про підтримку об'єктів Java і можливості їх використання як об'єктів PHP.

У ході виконання кваліфікаційної роботи – було з'ясовано, що головна область застосування PHP – написання скриптів, що працюють на стороні сервера; таким чином, мова PHP здатна виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або відсилати й приймати cookies. Але мова PHP здатна виконувати набагато більше призначень та функцій.

Мова PHP доступна для більшості операційних систем, включаючи Linux, багатьох модифікацій Unix (такі як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS та багато інших. Також у PHP включена підтримка більшості сучасних веб-серверів, таких як Apache, IIS і інших. В принципі, підійде будь-який веб-сервер, здатний використовувати бінарний файл FastCGI PHP, наприклад, lighttpd або nginx. PHP може працювати або в якості модуля або функціонувати як процесор CGI.

Також, після виконання кваліфікаційної роботи була досягнута її мета, що полягала у всебічному аналізі засобів створення сценарних додатків.

Отримані результати роботи:

а) виявлені нові закономірності впливу та застосування мови програмування PHP версії 5.40;

б) дістало подальший розвиток використання нових інструментів у процесі досліджень, а саме інструментів Notepad++, DzSoft PHP Editor та Scriptomania;

в) дістав подальший розвиток проведення самостійного експерименту трасування PHP коду.

Практичне значення – кваліфікаційна робота носить реальну значущість у галузі веб-розробок, що підтверджується виконаним аналізом засобів розробки та досліджень сценарних додатків.

В роботі розроблено методику створення сценарних додатків за допомогою необхідних програмних алгоритмів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lerdorf R., Tatroe K., Kaehms, B. Programming Php. Boston: O'Reilly Media Inc., 2002. 489 p.
2. Rehman R. U. Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID. Phoenix: Prentice Hall Professional, 2003. 108 p.
3. Великодний С. С., Тимофєєва О. С. Реінжиніринг програмного забезпечення інформаційних систем. Монографія. Одеса: Видавничий дім «Гельветика», 2020. 162 с.
4. Welling L., Thomson L. PHP et MySQL. New York: CampusPress, 2005. 230 p.
5. Великодний С. С. Моделі та методи проактивного управління проектами із розвитку програмних систем і продуктів. Монографія. Одеса: Одеський державний екологічний університет, 2021. 322 с. ISBN 978-966-186-182-3. (URL: <http://eprints.library.odku.edu.ua/id/eprint/9595/>)
6. Ullman L. PHP and MySQL for dynamic web sites: visual quickpro guide. Los Angeles: Peachpit Press, 2011. 496 p.
7. Tatroe K., MacIntyre P. Programming PHP: Creating dynamic web pages. Boston: O'Reilly Media, 2020. 428 p.
8. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання поведінкової частини. Вісник Кременчуцького національного університету ім. Михайла Остроградського. 2019. Вип. 2 (115). С. 117–126.
9. Gilmore W. J. Beginning PHP and MySQL: from novice to professional. San Francisco: Apress, 2010. 349 p.

10. Petukhin D., Velykodniy S., Kozlovskaya V. Modeling the space of possible states of the lesson schedule in higher education institutions. International Scientific and Practical Conference "Intellectual Systems and Information Technologies", 13-19 sep. 2021, Odesa, Ukraine. PP. 230–237.
11. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. Boston: O'Reilly Media Inc., 2014. 673 p.
12. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. Applied Aspects of Information Technology. 2019. Vol. 2. No 3. P. 186–205.
13. Suehring S., Valade J. PHP, MySQL, Javascript & HTML5 all-in-one for Dummies. New York: John Wiley & Sons, 2017. 695 p.
14. Великодний С. С. Моделювання складних процесів та систем (Частина 1): конспект лекцій. Одеса: Одеський державний екологічний університет, 2021. 92 с. ISBN 978-966-186-181-6. <http://eprints.library.odeku.edu.ua/id/eprint/9494/>
15. Stauffer M. Laravel: Up & running: A framework for building modern PHP Apps. Boston: O'Reilly Media, 2019. 530 p.
16. Velykodniy S., Burlachenko Zh., Zaitseva-Velykodna S. Modelling the behavioural component of the emergent parallel processes of working with graph databases using Petri net-tools. International Journal of Parallel, Emergent and Distributed Systems. (Scopus) 2021. Vol. 36. Iss. 6. P. 498-515. DOI: <https://doi.org/10.1080/17445760.2021.1934836>. Taylor & Francis Group, England & Wales. London.
17. Welling L., Thomson L. PHP et MySQL. Houston: CampusPress, 2005. 230 p.
18. Velykodniy S., Burlachenko Zh., Zaitseva-Velykodna S. Software for automated design of network graphics of software systems reengineering. Scientific Journal Herald of Advanced Information Technology. 2019. No 2 (03). P. 20–32.

19. Великодний С. С., Тимофєєва О. С. Спосіб мультилінгвістичного перекодування програмного забезпечення складних інформаційних систем та технологій. Наукові праці ОНАЗ ім. О. С. Попова. 2017. № 2. С. 153–159.
20. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. Вісник Кременчуцького національного університету ім. Михайла Остроградського. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI: 10.30929/1995-0519.2019.3.130-139.