ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп’ютерних

і телекомунікаційних систем

Кафедра інженерії програмного забезпечення

**Лабораторна робота № 5**

з дисципліни «Web-програмування» на тему:

Відображення документа на багатьох пристроях

**Виконав:**

Студент 3 курсу, групи ІПЗ-17, варіант № 15 . . С. В. Романчук

(підпис)

**Перевірив:**  . О.М. Яшина

(підпис)

Хмельницький – 2020 р.

**Тема**: Відображення документа на багатьох пристроях.

**Мета роботи**: Дослідити проблеми відображення web-сторінок на різних

пристроях та вивчити способи їх усунення.

Хід роботи

Завдання:

5.3.1 Ознайомитися з теоретичними відомостями,

необхідними для виконання роботи.

5.3.2 Змінити стилі сторінок, розроблених в лабораторній

роботі №4 таким чином, щоб вони стали зручними для

використання на пристроях з розмірами екрану від 320px до

1920px. Для перевірки відображення можна використовувати

вбудований емулятор інспектора коду Google Chrome.

4.3.3 Оформити звіт з роботи.

4.3.4 Відповісти на контрольні питання.

Код сторінки index.html

<!DOCTYPE html>

<html>

<head>

<title>PHP</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="./css/main.css" rel='stylesheet' type='text/css'>

</head>

<body>

<!-- Navigation Bar -->

<div class="navbar">

<a href="#" class="selected">Скорочена довідка</a>

<a href="./history.html">Історія</a>

<a href="./features.html">Особливості</a>

<a href="./syntax.html">Синтаксис</a>

<a href="./rating.html">Недоліки</a>

</div>

<!-- The flexible grid (content) -->

<div class="row">

<div class="side">

<h3>Зміст сторінки</h3>

<li><a href="index.html#phpIs">PHP це</a></li>

<li><a href="index.html#recognition">Визнання та поширення</a></li>

<li><a href="index.html#critics">Критика</a></li>

</div>

<div class="main">

<h1>PHP</h1>

<h2 id="phpIs">PHP це</h2>

<hr>

<div class="historyContent" >

<div class="historyContentFirstPart">

<p>PHP (Personal Home Page Tools) — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення.</p>

<p>PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.</p>

</div>

</div>

<h2 class="hTeg" id="recognition">Визнання та поширення</h2>

<hr>

<div class="historyContentSecondPart">

<p>PHP 4, що працює на цьому ядрі, вийшов 2000 року. На додаток до збільшення продуктивності, PHP 4 мав нові можливості щодо підтримки сесій, буферизацію виводу, безпечні способи обробки інформації, що вводиться користувачем, і нові мовні конструкції. З виходом 4 версії PHP став використовуватися вже на більш ніж 20 % доменів Інтернету.</p>

<p>Протягом 2000—2004 років продовжувалися активні роботи з покращення четвертої версії, але майже відразу PHP Group приступила до продумування можливостей нової версії. В першу чергу було вирішено підсилити об'єктні можливості мови, що дозволяло використовувати його для реалізації масштабних проектів. Роботи із створення п'ятої версії велися тривалий час, в них брало участь рекордна кількість фахівців, зокрема Стерлінг Хьюз (англ. Sterling Hughes) і Маркус Бергера (англ. Marcus Boerger).</p>

<p>У липні 2004 року виходить офіційний реліз PHP 5. В першу чергу, як і планувалося, було перероблено весь механізм роботи з об'єктами. І якщо в попередніх версіях об'єктно-орієнтоване програмування на PHP було можливе в мінімальному ступені, а тому і використовувалося на практиці не часто, то PHP 5 володіє прекрасним потенціалом реалізації об'єктного програмування. Окрім цього, PHP збагатився рядом цінних розширень для роботи з XML, різними джерелами даних, генерації графіки і інше.</p>

<p>Серед інших украй корисних доповнень в PHP 5 слід зазначити нову схему обробки винятків. Конструкція try/catch/throw дозволяє весь код обробки помилок локалізувати в одному місці сценарію.</p>

<p>Всі основні бібліотеки для роботи з XML, запозичені в PHP 4, були піддані серйозній переробці. Такі популярні розширення, як SAX, DOM і XSLT, тепер використовують інструмент libxml2, що робить їх ще ефективнішими.</p>

<p>У PHP 5 також включені два нові модулі для роботи з протоколами — SimpleXML і 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, що обмінюються інформацією з іншими застосунками за допомогою XML-повідомлень поверх існуючих веб-протоколів, наприклад HTTP. Модуль для роботи з SOAP для PHP 5 надає розробникам засіб для достатньо швидкого створення ефективних SOAP-клієнтів і SOAP-серверів.</p>

<p>Новий модуль PHP 5 MySQLi (MySQL Improved) призначений для роботи з 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, контроль транзакцій, підтримка реплікації та інші.</p>

<p>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.</p>

<p>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.</p>

<p>3 грудня 2015 року було оголошено про вихід PHP версії 7.0.0.</p>

<p>Нова версія ґрунтується на експериментальній гілці PHP, яка спочатку називалася phpng (PHP Next Generation — наступне покоління), і розроблялася з нахилом 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.</p>

</div>

<h2 class="hSecondTeg" id="critics">Критика</h2>

<hr>

<div class="historyContentSecondPart">

<p>У статті «PHP: a fractal of bad design» представлений докладний і ґрунтовний огляд проблем в дизайні мови програмування PHP. Автор показує винятковість PHP як одної з найбільш неопрацьованих мов, проблеми в якій мають систематичний характер. Зокрема у статті продемонстровані проблеми в самій мові, бібліотеці функцій, структурах, механізмах роботи з даними, екосистемі, засобах зневадження. Вказані недоробки в безпеці, надійності, цілісності та передбачуваності.</p>

<p>Якщо розглядати безпеку, то як приклад спочатку порочної практики наводиться розрізненість засобів для чищення та нормалізації даних перед їхнім використанням у різних операціях, що є прекрасною підмогою для виникнення в застосунках вразливостей, що дозволяють здійснити підстановку SQL-коду або вбудовування JavaScript на сторінку. Згадки також заслуговує підхід «небезпечний за умовчанням», який тільки останнім часом став переглядатися розробниками PHP, наприклад, причиною безлічі вразливостей є використання register\_globals і підтримка виконання зовнішнього коду за URL в директиві include. З проблем у самому інтерпретаторі відзначається спроба виправити у 2007 році цілочисельну вразливість через перевірку "if (size > INT\_MAX) return NULL;"; помилка в реалізації функції crypt() в PHP 5.3.7 через яку можна було зайти з будь-яким паролем; DoS-уразливість в PHP 5.4, пов'язана з виділенням пам'яті на підставі переданого користувачем значення в HTTP-заголовку Content-Length.</p>

</div>

</div>

</div>

<!-- Footer -->

<div class="footer">

<div class="footerContainer">

<h2>Відомості про автора</h2>

<span>Лабораторна робота розроблена: <a href="C:\Users\N9\Desktop\3 курс 2 семестр\WEB-програмування\Lab3\resume.html">Романчук Сергій ІПЗ-17-1</a></span>

</div>

<p id="quote"></p>

</div>

</body>

</html>

Код сторінки features.html

<!DOCTYPE html>

<html>

<head>

<title>PHP</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="./css/history.css" rel='stylesheet' type='text/css'>

</head>

<body>

<!-- Navigation Bar -->

<div class="navbar">

<a href="./index.html">Скорочена довідка</a>

<a href="#" class="selected">Історія</a>

<a href="./features.html">Особливості</a>

<a href="./syntax.html">Синтаксис</a>

<a href="./rating.html">Недоліки</a>

</div>

<!-- The flexible grid (content) -->

<div class="row">

<div class="side">

<h3>Довідка</h3>

<li><a href="history.html#firstVersions">Перші версії</a></li>

<li><a href="history.html#php3AndPhp4">PHP 3 та PHP 4</a></li>

<li><a href="history.html#php5">PHP 5</a></li>

<li><a href="history.html#php6">PHP 6 та Unicode</a></li>

<li><a href="history.html#php7">PHP 7</a></li>

</div>

<div class="main">

<h1>PHP</h1>

<h2 id="firstVersions">Перші версії</h2> <!-- history -->

<hr>

<div class="firstVersions" > <!-- historyContent -->

<div class="historyContentFirstPart"> <!-- historyContentFirstPart -->

<p>Історія PHP починається з 1995 року, коли Расмус Лердорф (англ. Rasmus Lerdorf) створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на веб-сайті. Потім, коли цим застосунком вже користувалися кілька осіб, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Інструменти для особистої домашньої сторінки (Personal Home Page Tools) версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.</p>

<p>Як це завжди буває, терміново було потрібне доопрацювання і нові доповнення. Для їхньої реалізації Расмус створює нову версію пакету, тепер уже написану на С. Отриманий таким чином інструмент набуває робочої назви PHP/FI (укр. Персональна Домашня сторінка / Інтерпретатор Форм, англ. Personal Home Page / Forms Interpreter), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, можливість вбудовування PHP операторів в html-код сторінки, автоматичну інтерпретацію форм, інтеграцію з базами даних. При цьому все працювало досить швидко, оскільки PHP прикомпілювалася до веб-сервера Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1 % всіх веб-серверів).</p>

<p>Того ж 1997 року до проекту PHP підключилися Зев Сураскі (англ. Zeev Suraski) і Енді Ґутманс (англ. Andi Gutmans). Ці студенти Техніону, одного з найкращих ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зев і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням.</p>

<p>Закінчивши роботу, Зев і Енді домовились з Расмусом про співпрацю в галузі розвитку та вдосконалення мови. З цієї миті з'являється PHP Group — група однодумців, що працюють над розвитком технології PHP. Одержаний продукт з'явився на світ 1998 року під назвою PHP 3.</p>

<p>При цьому головною особливість PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 року кількість користувачів PHP перевищила 100 тисяч, а PHP був уже встановлений на понад 10 % серверах Інтернету. Водночас значному поширенню даної мови сприяли публікації в електронній пресі та видання посібників із PHP.</p>

<p>Відразу ж після виходу PHP 3, Енді Гутманс і Зев Сураскі почали переробку ядра PHP. В першу чергу належало розв'язати проблему підвищення продуктивності. Новий продукт, названий Zend Engine (від імен творців: Zeev і Andi), успішно справлявся з поставленим завданням і був реалізований 1999 року. Основними реалізованими ідеями є можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.</p>

</div>

</div>

<h2 class="hTeg" id="php3AndPhp4">PHP 3 та PHP 4</h2>

<hr>

<div class="historyContentSecondPart">

<p>Зеєв Сураскі та Ані Гутманс переробили аналізатор в 1997 році і сформували базу PHP 3, змінивши назву мови на рекурсивний акронім PHP: Hypertext Preprocessor. Після цього почалося публічне тестування PHP 3, а офіційний запуск відбувся в червні 1998 року. Після цього Сураскі та Гутманс розпочали нове перекодування ядра PHP, видавши Zend Engine у 1999 році. Вони також заснували Zend Technologies в Рамат-Гані, Ізраїль.</p>

<p>22 травня 2000 р. Був випущений PHP 4 з підтримкою Zend Engine 1.0. Станом на серпень 2008 року ця філія досягла версії 4.4.9. PHP 4 більше не розробляється, оновлення безпеки також більше не були випущені.</p>

</div>

<h2 class="hSecondTeg" id="php5">PHP 5</h2>

<hr>

<div class="historyContentSecondPart">

<p>14 липня 2004 р. Був випущений PHP 5 з новим двигуном Zend Engine II. PHP 5 включав нові функції, такі як покращена підтримка об'єктно-орієнтованого програмування, розширення PHP Data Objects (PDO) (який містить легкий і послідовний інтерфейс для доступу до баз даних) та численні поліпшення продуктивності. У 2008 році PHP 5 став єдиною стабільною версією, що розроблялася. Пізній статичний зв'язок відсутній у PHP і був доданий у версії 5.3.</p>

<p>Багато високопрофесійних проектів з відкритим кодом з 5 лютого 2008 року перестали підтримувати PHP 4 в новому коді, оскільки це ініціатива GoPHP5, що надається консорціумом розробників PHP, що сприяє переходу від PHP 4 до PHP 5.</p>

<p>З часом перекладачі PHP стали доступними для більшості існуючих 32-розрядних та 64-розрядних операційних систем, або будували їх з вихідного коду PHP, або використовували попередньо побудовані двонарні файли. Для версій PHP версії 5.3 та 5.4 єдиними доступними двосторонніми дистрибутивами Microsoft Windows були 32-розрядні версії x86, що вимагають 32-розрядний режим сумісності Windows при використанні інформаційних служб Інтернету (IIS) на 64-розрядній платформі Windows. PHP версії 5.5 зробив збірки 64-розрядних x86-64 доступними для Microsoft Windows.</p>

</div>

<h2 class="hThirtyTeg" id="php6">PHP 6 та Unicode</h2>

<hr>

<div class="historyContentSecondPart">

<p>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.</p>

<p>Проте дефіцит розробників, які зрозуміли необхідні зміни та проблеми продуктивності, що виникають внаслідок перетворення на UTF-16 та з нього, що рідко використовується в веб-контексті, призвело до затримок проекту. Як результат, випуск PHP 5.3 був створений у 2009 році, при цьому багато не-Unicode-функцій було відновлено з PHP 6, зокрема простору імен. У березні 2010 року проект у своїй нинішній формі був офіційно відкинутий, і був підготовлений випуск PHP 5.4, що містить більшість опублікованих функцій, що не входять до Unicode, з PHP 6, такі як риси та переприв'язка до закриття. Початкові сподівання полягали в тому, що для інтеграції з Unicode був би сформований новий план, але з 2014 року ніхто не був прийнятий.</p>

</div>

<h2 class="hFourthTeg" id="php7">PHP 7</h2>

<hr>

<div class="historyContentSecondPart">

<p>Протягом 2014 та 2015 років було розроблено нову основну версію PHP, яку було пронумеровано PHP 7. Нумерація цієї версії викликала дебати. Хоча реліз PHP 6 з Unicode ніколи не був випущений, декілька назв статей і книг посилалися на назву PHP 6, що могло викликати плутанину, якби новий реліз повторно використовував цю назву. Після голосування було обрано назву PHP 7</p>

</div>

</div>

</div>

<!-- Footer -->

<div class="footer">

<div class="footerContainer">

<h2>Відомості про автора</h2>

<span>Лабораторна робота розроблена: <a href="C:\Users\N9\Desktop\3 курс 2 семестр\WEB-програмування\Lab3\resume.html">Романчук Сергій ІПЗ-17-1</a></span>

</div>

<p id="quote"></p>

</div>

</body>

</html>

Код сторінки syntax.html

<!DOCTYPE html>

<html>

<head>

<title>PHP</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="./css/history.css" rel='stylesheet' type='text/css'>

</head>

<body>

<!-- Navigation Bar -->

<div class="navbar">

<a href="./index.html">Скорочена довідка</a>

<a href="#" class="selected">Історія</a>

<a href="./features.html">Особливості</a>

<a href="./syntax.html">Синтаксис</a>

<a href="./rating.html">Недоліки</a>

</div>

<!-- The flexible grid (content) -->

<div class="row">

<div class="side">

<h3>Довідка</h3>

<li><a href="history.html#firstVersions">Перші версії</a></li>

<li><a href="history.html#php3AndPhp4">PHP 3 та PHP 4</a></li>

<li><a href="history.html#php5">PHP 5</a></li>

<li><a href="history.html#php6">PHP 6 та Unicode</a></li>

<li><a href="history.html#php7">PHP 7</a></li>

</div>

<div class="main">

<h1>PHP</h1>

<h2 id="firstVersions">Перші версії</h2> <!-- history -->

<hr>

<div class="firstVersions" > <!-- historyContent -->

<div class="historyContentFirstPart"> <!-- historyContentFirstPart -->

<p>Історія PHP починається з 1995 року, коли Расмус Лердорф (англ. Rasmus Lerdorf) створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на веб-сайті. Потім, коли цим застосунком вже користувалися кілька осіб, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Інструменти для особистої домашньої сторінки (Personal Home Page Tools) версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.</p>

<p>Як це завжди буває, терміново було потрібне доопрацювання і нові доповнення. Для їхньої реалізації Расмус створює нову версію пакету, тепер уже написану на С. Отриманий таким чином інструмент набуває робочої назви PHP/FI (укр. Персональна Домашня сторінка / Інтерпретатор Форм, англ. Personal Home Page / Forms Interpreter), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, можливість вбудовування PHP операторів в html-код сторінки, автоматичну інтерпретацію форм, інтеграцію з базами даних. При цьому все працювало досить швидко, оскільки PHP прикомпілювалася до веб-сервера Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1 % всіх веб-серверів).</p>

<p>Того ж 1997 року до проекту PHP підключилися Зев Сураскі (англ. Zeev Suraski) і Енді Ґутманс (англ. Andi Gutmans). Ці студенти Техніону, одного з найкращих ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зев і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням.</p>

<p>Закінчивши роботу, Зев і Енді домовились з Расмусом про співпрацю в галузі розвитку та вдосконалення мови. З цієї миті з'являється PHP Group — група однодумців, що працюють над розвитком технології PHP. Одержаний продукт з'явився на світ 1998 року під назвою PHP 3.</p>

<p>При цьому головною особливість PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 року кількість користувачів PHP перевищила 100 тисяч, а PHP був уже встановлений на понад 10 % серверах Інтернету. Водночас значному поширенню даної мови сприяли публікації в електронній пресі та видання посібників із PHP.</p>

<p>Відразу ж після виходу PHP 3, Енді Гутманс і Зев Сураскі почали переробку ядра PHP. В першу чергу належало розв'язати проблему підвищення продуктивності. Новий продукт, названий Zend Engine (від імен творців: Zeev і Andi), успішно справлявся з поставленим завданням і був реалізований 1999 року. Основними реалізованими ідеями є можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.</p>

</div>

</div>

<h2 class="hTeg" id="php3AndPhp4">PHP 3 та PHP 4</h2>

<hr>

<div class="historyContentSecondPart">

<p>Зеєв Сураскі та Ані Гутманс переробили аналізатор в 1997 році і сформували базу PHP 3, змінивши назву мови на рекурсивний акронім PHP: Hypertext Preprocessor. Після цього почалося публічне тестування PHP 3, а офіційний запуск відбувся в червні 1998 року. Після цього Сураскі та Гутманс розпочали нове перекодування ядра PHP, видавши Zend Engine у 1999 році. Вони також заснували Zend Technologies в Рамат-Гані, Ізраїль.</p>

<p>22 травня 2000 р. Був випущений PHP 4 з підтримкою Zend Engine 1.0. Станом на серпень 2008 року ця філія досягла версії 4.4.9. PHP 4 більше не розробляється, оновлення безпеки також більше не були випущені.</p>

</div>

<h2 class="hSecondTeg" id="php5">PHP 5</h2>

<hr>

<div class="historyContentSecondPart">

<p>14 липня 2004 р. Був випущений PHP 5 з новим двигуном Zend Engine II. PHP 5 включав нові функції, такі як покращена підтримка об'єктно-орієнтованого програмування, розширення PHP Data Objects (PDO) (який містить легкий і послідовний інтерфейс для доступу до баз даних) та численні поліпшення продуктивності. У 2008 році PHP 5 став єдиною стабільною версією, що розроблялася. Пізній статичний зв'язок відсутній у PHP і був доданий у версії 5.3.</p>

<p>Багато високопрофесійних проектів з відкритим кодом з 5 лютого 2008 року перестали підтримувати PHP 4 в новому коді, оскільки це ініціатива GoPHP5, що надається консорціумом розробників PHP, що сприяє переходу від PHP 4 до PHP 5.</p>

<p>З часом перекладачі PHP стали доступними для більшості існуючих 32-розрядних та 64-розрядних операційних систем, або будували їх з вихідного коду PHP, або використовували попередньо побудовані двонарні файли. Для версій PHP версії 5.3 та 5.4 єдиними доступними двосторонніми дистрибутивами Microsoft Windows були 32-розрядні версії x86, що вимагають 32-розрядний режим сумісності Windows при використанні інформаційних служб Інтернету (IIS) на 64-розрядній платформі Windows. PHP версії 5.5 зробив збірки 64-розрядних x86-64 доступними для Microsoft Windows.</p>

</div>

<h2 class="hThirtyTeg" id="php6">PHP 6 та Unicode</h2>

<hr>

<div class="historyContentSecondPart">

<p>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.</p>

<p>Проте дефіцит розробників, які зрозуміли необхідні зміни та проблеми продуктивності, що виникають внаслідок перетворення на UTF-16 та з нього, що рідко використовується в веб-контексті, призвело до затримок проекту. Як результат, випуск PHP 5.3 був створений у 2009 році, при цьому багато не-Unicode-функцій було відновлено з PHP 6, зокрема простору імен. У березні 2010 року проект у своїй нинішній формі був офіційно відкинутий, і був підготовлений випуск PHP 5.4, що містить більшість опублікованих функцій, що не входять до Unicode, з PHP 6, такі як риси та переприв'язка до закриття. Початкові сподівання полягали в тому, що для інтеграції з Unicode був би сформований новий план, але з 2014 року ніхто не був прийнятий.</p>

</div>

<h2 class="hFourthTeg" id="php7">PHP 7</h2>

<hr>

<div class="historyContentSecondPart">

<p>Протягом 2014 та 2015 років було розроблено нову основну версію PHP, яку було пронумеровано PHP 7. Нумерація цієї версії викликала дебати. Хоча реліз PHP 6 з Unicode ніколи не був випущений, декілька назв статей і книг посилалися на назву PHP 6, що могло викликати плутанину, якби новий реліз повторно використовував цю назву. Після голосування було обрано назву PHP 7</p>

</div>

</div>

</div>

<!-- Footer -->

<div class="footer">

<div class="footerContainer">

<h2>Відомості про автора</h2>

<span>Лабораторна робота розроблена: <a href="C:\Users\N9\Desktop\3 курс 2 семестр\WEB-програмування\Lab3\resume.html">Романчук Сергій ІПЗ-17-1</a></span>

</div>

<p id="quote"></p>

</div>

</body>

</html>

Код сторінки rating.html

<!DOCTYPE html>

<html>

<head>

<title>PHP</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="./css/rating.css" rel='stylesheet' type='text/css'>

</head>

<body>

<div class="navbar">

<a href="./index.html">Скорочена довідка</a>

<a href="./history.html">Історія</a>

<a href="./features.html">Особливості</a>

<a href="./syntax.html">Синтаксис</a>

<a href="#" class="selected">Недоліки</a>

</div>

<!-- The flexible grid (content) -->

<div class="row">

<div class="side">

<h3>Довідка</h3>

<li><a href="rating.html#version">Недоліки мови PHP</a></li>

</div>

<div class="main">

<h1>PHP</h1>

<h2 class="hTeg" id="version">Недоліки мови PHP</h2>

<hr>

<div class="historyContentSecondPart">

<ol>

<li>Незручність дизайну мови</li>

<li>Змінні з символом <i>$</i></li>

<li>Складні назви поширених функцій (html\_entity\_decode, Шаблон:Php, nl2br тощо)</li>

<li>Не підтримується Unicode у версіях до 6.0</li>

<li>Непередбачуваність нових версій PHP.</li>

</ol>

</div>

</div>

</div>

<!-- Footer -->

<div class="footer">

<div class="footerContainer">

<h2>Відомості про автора</h2>

<span>Лабораторна робота розроблена: <a href="C:\Users\N9\Desktop\3 курс 2 семестр\WEB-програмування\Lab3\resume.html">Романчук Сергій ІПЗ-17-1</a></span>

</div>

<p id="quote"></p>

</div>

</body>

</html>

Код сторінки history.html

<!DOCTYPE html>

<html>

<head>

<title>PHP</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="./css/history.css" rel='stylesheet' type='text/css'>

</head>

<body>

<!-- Navigation Bar -->

<div class="navbar">

<a href="./index.html">Скорочена довідка</a>

<a href="#" class="selected">Історія</a>

<a href="./features.html">Особливості</a>

<a href="./syntax.html">Синтаксис</a>

<a href="./rating.html">Недоліки</a>

</div>

<!-- The flexible grid (content) -->

<div class="row">

<div class="side">

<h3>Довідка</h3>

<li><a href="history.html#firstVersions">Перші версії</a></li>

<li><a href="history.html#php3AndPhp4">PHP 3 та PHP 4</a></li>

<li><a href="history.html#php5">PHP 5</a></li>

<li><a href="history.html#php6">PHP 6 та Unicode</a></li>

<li><a href="history.html#php7">PHP 7</a></li>

</div>

<div class="main">

<h1>PHP</h1>

<h2 id="firstVersions">Перші версії</h2>

<hr>

<div class="firstVersions" >

<div class="historyContentFirstPart">

<p>Історія PHP починається з 1995 року, коли Расмус Лердорф (англ. Rasmus Lerdorf) створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на веб-сайті. Потім, коли цим застосунком вже користувалися кілька осіб, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Інструменти для особистої домашньої сторінки (Personal Home Page Tools) версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.</p>

<p>Як це завжди буває, терміново було потрібне доопрацювання і нові доповнення. Для їхньої реалізації Расмус створює нову версію пакету, тепер уже написану на С. Отриманий таким чином інструмент набуває робочої назви PHP/FI (укр. Персональна Домашня сторінка / Інтерпретатор Форм, англ. Personal Home Page / Forms Interpreter), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, можливість вбудовування PHP операторів в html-код сторінки, автоматичну інтерпретацію форм, інтеграцію з базами даних. При цьому все працювало досить швидко, оскільки PHP прикомпілювалася до веб-сервера Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1 % всіх веб-серверів).</p>

<p>Того ж 1997 року до проекту PHP підключилися Зев Сураскі (англ. Zeev Suraski) і Енді Ґутманс (англ. Andi Gutmans). Ці студенти Техніону, одного з найкращих ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зев і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням.</p>

<p>Закінчивши роботу, Зев і Енді домовились з Расмусом про співпрацю в галузі розвитку та вдосконалення мови. З цієї миті з'являється PHP Group — група однодумців, що працюють над розвитком технології PHP. Одержаний продукт з'явився на світ 1998 року під назвою PHP 3.</p>

<p>При цьому головною особливість PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 року кількість користувачів PHP перевищила 100 тисяч, а PHP був уже встановлений на понад 10 % серверах Інтернету. Водночас значному поширенню даної мови сприяли публікації в електронній пресі та видання посібників із PHP.</p>

<p>Відразу ж після виходу PHP 3, Енді Гутманс і Зев Сураскі почали переробку ядра PHP. В першу чергу належало розв'язати проблему підвищення продуктивності. Новий продукт, названий Zend Engine (від імен творців: Zeev і Andi), успішно справлявся з поставленим завданням і був реалізований 1999 року. Основними реалізованими ідеями є можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.</p>

</div>

</div>

<h2 class="hTeg" id="php3AndPhp4">PHP 3 та PHP 4</h2>

<hr>

<div class="historyContentSecondPart">

<p>Зеєв Сураскі та Ані Гутманс переробили аналізатор в 1997 році і сформували базу PHP 3, змінивши назву мови на рекурсивний акронім PHP: Hypertext Preprocessor. Після цього почалося публічне тестування PHP 3, а офіційний запуск відбувся в червні 1998 року. Після цього Сураскі та Гутманс розпочали нове перекодування ядра PHP, видавши Zend Engine у 1999 році. Вони також заснували Zend Technologies в Рамат-Гані, Ізраїль.</p>

<p>22 травня 2000 р. Був випущений PHP 4 з підтримкою Zend Engine 1.0. Станом на серпень 2008 року ця філія досягла версії 4.4.9. PHP 4 більше не розробляється, оновлення безпеки також більше не були випущені.</p>

</div>

<h2 class="hSecondTeg" id="php5">PHP 5</h2>

<hr>

<div class="historyContentSecondPart">

<p>14 липня 2004 р. Був випущений PHP 5 з новим двигуном Zend Engine II. PHP 5 включав нові функції, такі як покращена підтримка об'єктно-орієнтованого програмування, розширення PHP Data Objects (PDO) (який містить легкий і послідовний інтерфейс для доступу до баз даних) та численні поліпшення продуктивності. У 2008 році PHP 5 став єдиною стабільною версією, що розроблялася. Пізній статичний зв'язок відсутній у PHP і був доданий у версії 5.3.</p>

<p>Багато високопрофесійних проектів з відкритим кодом з 5 лютого 2008 року перестали підтримувати PHP 4 в новому коді, оскільки це ініціатива GoPHP5, що надається консорціумом розробників PHP, що сприяє переходу від PHP 4 до PHP 5.</p>

<p>З часом перекладачі PHP стали доступними для більшості існуючих 32-розрядних та 64-розрядних операційних систем, або будували їх з вихідного коду PHP, або використовували попередньо побудовані двонарні файли. Для версій PHP версії 5.3 та 5.4 єдиними доступними двосторонніми дистрибутивами Microsoft Windows були 32-розрядні версії x86, що вимагають 32-розрядний режим сумісності Windows при використанні інформаційних служб Інтернету (IIS) на 64-розрядній платформі Windows. PHP версії 5.5 зробив збірки 64-розрядних x86-64 доступними для Microsoft Windows.</p>

</div>

<h2 class="hThirtyTeg" id="php6">PHP 6 та Unicode</h2>

<hr>

<div class="historyContentSecondPart">

<p>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.</p>

<p>Проте дефіцит розробників, які зрозуміли необхідні зміни та проблеми продуктивності, що виникають внаслідок перетворення на UTF-16 та з нього, що рідко використовується в веб-контексті, призвело до затримок проекту. Як результат, випуск PHP 5.3 був створений у 2009 році, при цьому багато не-Unicode-функцій було відновлено з PHP 6, зокрема простору імен. У березні 2010 року проект у своїй нинішній формі був офіційно відкинутий, і був підготовлений випуск PHP 5.4, що містить більшість опублікованих функцій, що не входять до Unicode, з PHP 6, такі як риси та переприв'язка до закриття. Початкові сподівання полягали в тому, що для інтеграції з Unicode був би сформований новий план, але з 2014 року ніхто не був прийнятий.</p>

</div>

<h2 class="hFourthTeg" id="php7">PHP 7</h2>

<hr>

<div class="historyContentSecondPart">

<p>Протягом 2014 та 2015 років було розроблено нову основну версію PHP, яку було пронумеровано PHP 7. Нумерація цієї версії викликала дебати. Хоча реліз PHP 6 з Unicode ніколи не був випущений, декілька назв статей і книг посилалися на назву PHP 6, що могло викликати плутанину, якби новий реліз повторно використовував цю назву. Після голосування було обрано назву PHP 7</p>

</div>

</div>

</div>

<!-- Footer -->

<div class="footer">

<div class="footerContainer">

<h2>Відомості про автора</h2>

<span>Лабораторна робота розроблена: <a href="C:\Users\N9\Desktop\3 курс 2 семестр\WEB-програмування\Lab3\resume.html">Романчук Сергій ІПЗ-17-1</a></span>

</div>

<p id="quote"></p>

</div>

</body>

</html>

Код сторінки main.css

\* {

box-sizing: border-box;

}

/\* Style the body \*/

body {

font-family: Arial;

margin: 0;

}

/\* Header/logo Title \*/

.header {

padding: 10px;

text-align: center;

background: #2846a8;

color: #7181b6;

}

/\* Style the top navigation bar \*/

.navbar {

display: flex;

background-color: #333;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

.selected {

background-color: #ddd;

color: black !important;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 1%;

background-color: #6d6eb1;

padding: 20px;

height: 1800px;

}

.side a:link{

color: #def1ff;

}

.side a:hover{

color: #ddf1fa;

}

.side a:visited{

color: #9daefe;

}

.side h3 {

display: flex;

justify-content: left;

color: #def1ff;

}

.side ol li a {

text-decoration: none;

}

.side ol li {

padding: 5px 0;

}

/\* Main column \*/

.main {

font-size: 20px;

flex: 70%;

background-color: white;

padding: 20px;

}

/\* Fake image, just for this example \*/

.fakeimg {

background-color: #aaa;

width: 100%;

padding: 20px;

}

/\* Footer \*/

.footer {

padding: 10px;

text-align: center;

background: #c2dfff;

}

.footerContainer a {

font-size: 20px;

padding: 10px;

background: #49569c;

color: white;

text-decoration: none;

}

@media screen and (max-width: 700px) {

.row, .navbar, .navbar1, .footer {

flex-direction: column;

}

}

Код сторінки features.css

\* {

box-sizing: border-box;

}

/\* Style the body \*/

body {

font-family: Arial;

margin: 0;

}

/\* Header/logo Title \*/

.header {

padding: 10px;

text-align: center;

background: #2846a8;

color: #7181b6;

}

/\* Style the top navigation bar \*/

.navbar {

display: flex;

background-color: #333;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

.selected {

background-color: #ddd;

color: black !important;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 1%;

background-color: #6d6eb1;

padding: 20px;

height: 1400px;

}

.side h3 {

display: flex;

justify-content: left;

color: #def1ff;

}

.side a:link{

color: #def1ff;

}

.side a:hover{

color: #ddf1fa;

}

.side a:visited{

color: #9daefe;

}

.side ol li a {

text-decoration: none;

}

.side ol li {

padding: 5px 0;

}

/\* Main column \*/

.main {

font-size: 20px;

flex: 70%;

background-color: white;

padding: 20px;

}

.historyContentSecondPart ul li {

margin-top: 10px;

margin-left: 20px;

font-size: 20px;

list-style-type: none;

}

/\* Fake image, just for this example \*/

.fakeimg {

background-color: #aaa;

width: 100%;

padding: 20px;

}

/\* Footer \*/

.footer {

padding: 10px;

text-align: center;

background: #c2dfff;

}

.footerContainer a {

font-size: 20px;

padding: 10px;

background: #49569c;

color: white;

text-decoration: none;

}

/\* Responsive layout - when the screen is less than 700px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 700px) {

.row, .navbar, .navbar1 {

flex-direction: column;

}

}

Код сторінки rating.css

\* {

box-sizing: border-box;

}

/\* Style the body \*/

body {

font-family: Arial;

margin: 0;

}

/\* Header/logo Title \*/

.header {

padding: 10px;

text-align: center;

background: #2846a8;

color: #7181b6;

}

.navbar1 {

display: flex;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar1 a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar1 a:hover {

background-color: #ddd;

color: #5a6485;

}

/\* Style the top navigation bar \*/

.navbar {

display: flex;

background-color: #333;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

.selected {

background-color: #ddd;

color: black !important;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 1%;

background-color: #6d6eb1;

padding: 20px;

height: 1300px;

}

.side a:link{

color: #def1ff;

}

.side a:hover{

color: #ddf1fa;

}

.side a:visited{

color: #9daefe;

}

.side h3 {

display: flex;

justify-content: left;

color: #def1ff;

}

.side ol li a {

text-decoration: none;

}

.side ol li {

padding: 5px 0;

}

/\* Main column \*/

.main {

font-size: 20px;

flex: 70%;

background-color: white;

padding: 20px;

}

.historyContentFirstPart ol li {

padding: 10px 0;

}

.historyContentSecondPart ol li {

padding: 10px 0;

}

/\* Fake image, just for this example \*/

.fakeimg {

background-color: #aaa;

width: 100%;

padding: 20px;

}

/\* Footer \*/

.footer {

padding: 10px;

text-align: center;

background: #c2dfff;

}

.footerContainer a {

font-size: 20px;

padding: 10px;

background: #49569c;

color: white;

text-decoration: none;

}

/\* Responsive layout - when the screen is less than 700px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 700px) {

.row, .navbar, .navbar1 {

flex-direction: column;

}

}

Код сторінки syntax.css

\* {

box-sizing: border-box;

}

/\* Style the body \*/

body {

font-family: Arial;

margin: 0;

}

/\* Header/logo Title \*/

.header {

padding: 10px;

text-align: center;

background: #2846a8;

color: #7181b6;

}

/\* Style the top navigation bar \*/

.navbar {

display: flex;

background-color: #333;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

.selected {

background-color: #ddd;

color: black !important;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 1%;

background-color: #6d6eb1;

padding: 20px;

height: 800px;

}

.side a:link{

color: #def1ff;

}

.side a:hover{

color: #ddf1fa;

}

.side a:visited{

color: #9daefe;

}

.side h3 {

display: flex;

justify-content: left;

color: #def1ff;

}

.side ol li a {

text-decoration: none;

}

.side ol li {

padding: 5px 0;

}

/\* Main column \*/

.main {

font-size: 20px;

flex: 70%;

background-color: white;

padding: 20px;

}

.historyContentSecondPart ul li {

margin-top: 10px;

margin-left: 20px;

font-size: 20px;

list-style-type: none;

}

/\* Fake image, just for this example \*/

.fakeimg {

background-color: #aaa;

width: 100%;

padding: 20px;

}

/\* Footer \*/

.footer {

padding: 10px;

text-align: center;

background: #c2dfff;

}

.footerContainer a {

font-size: 20px;

padding: 10px;

background: #49569c;

color: white;

text-decoration: none;

}

/\* Responsive layout - when the screen is less than 700px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 700px) {

.row, .navbar, .navbar1 {

flex-direction: column;

}

}

Код сторінки history.css

\* {

box-sizing: border-box;

}

/\* Style the body \*/

body {

font-family: Arial;

margin: 0;

}

/\* Header/logo Title \*/

.header {

padding: 10px;

text-align: center;

background: #2846a8;

color: #7181b6;

}

/\* Style the top navigation bar \*/

.navbar {

display: flex;

background-color: #333;

justify-content: left;

}

/\* Style the navigation bar links \*/

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

.selected {

background-color: #ddd;

color: black !important;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 1%;

background-color: #6d6eb1;

padding: 20px;

height: 2000px;

}

.side a:link{

color: #def1ff;

}

.side a:hover{

color: #ddf1fa;

}

.side a:visited{

color: #9daefe;

}

.side h3 {

display: flex;

justify-content: left;

color: #def1ff;

}

.side ol li a {

text-decoration: none;

}

.side ol li {

padding: 5px 0;

}

/\* Main column \*/

.main {

font-size: 20px;

flex: 70%;

background-color: white;

padding: 20px;

}

.historyContentSecondPart ul li {

margin-top: 10px;

margin-left: 20px;

font-size: 20px;

list-style-type: none;

}

/\* Fake image, just for this example \*/

.fakeimg {

background-color: #aaa;

width: 100%;

padding: 20px;

}

/\* Footer \*/

.footer {

padding: 10px;

text-align: center;

background: #c2dfff;

}

.footerContainer a {

font-size: 20px;

padding: 10px;

background: #49569c;

color: white;

text-decoration: none;

}

/\* Responsive layout - when the screen is less than 700px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 700px) {

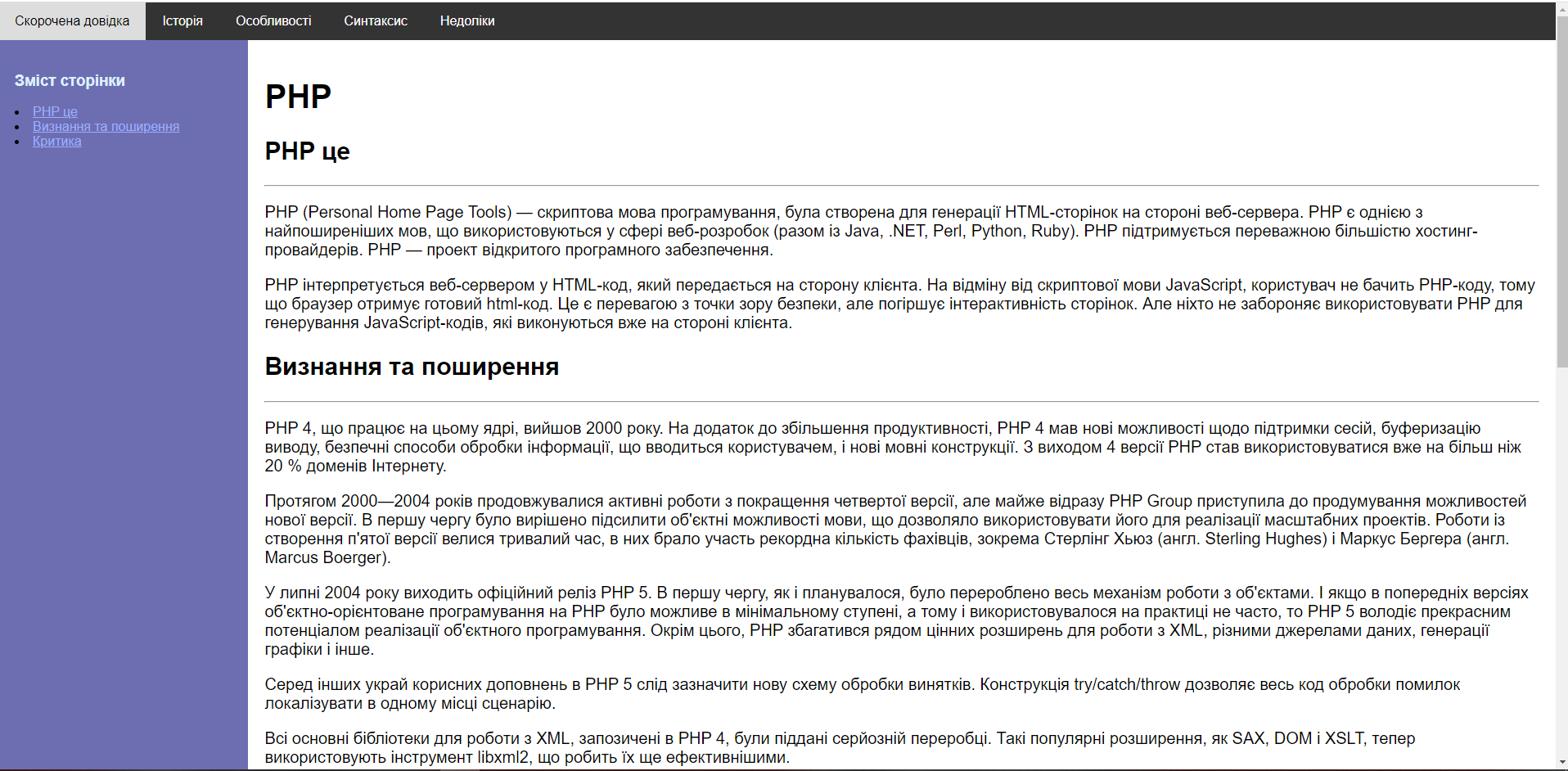
.row, .navbar, .navbar1 {

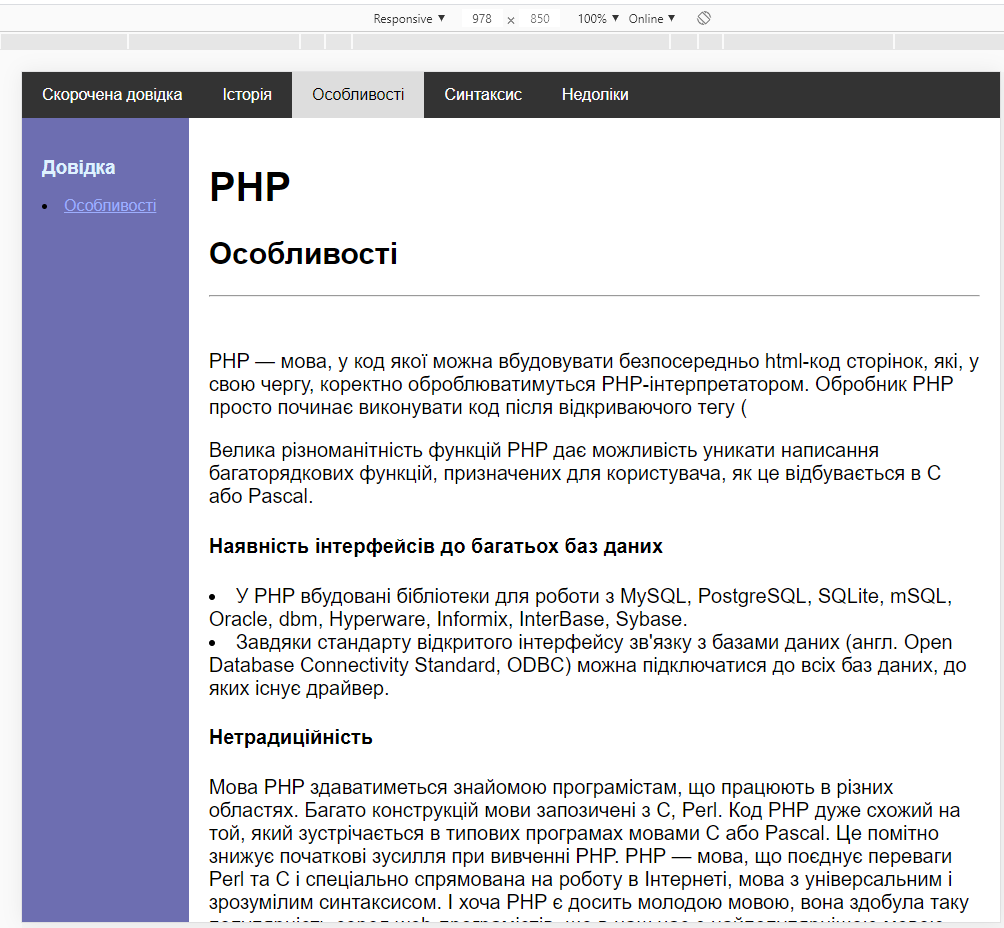
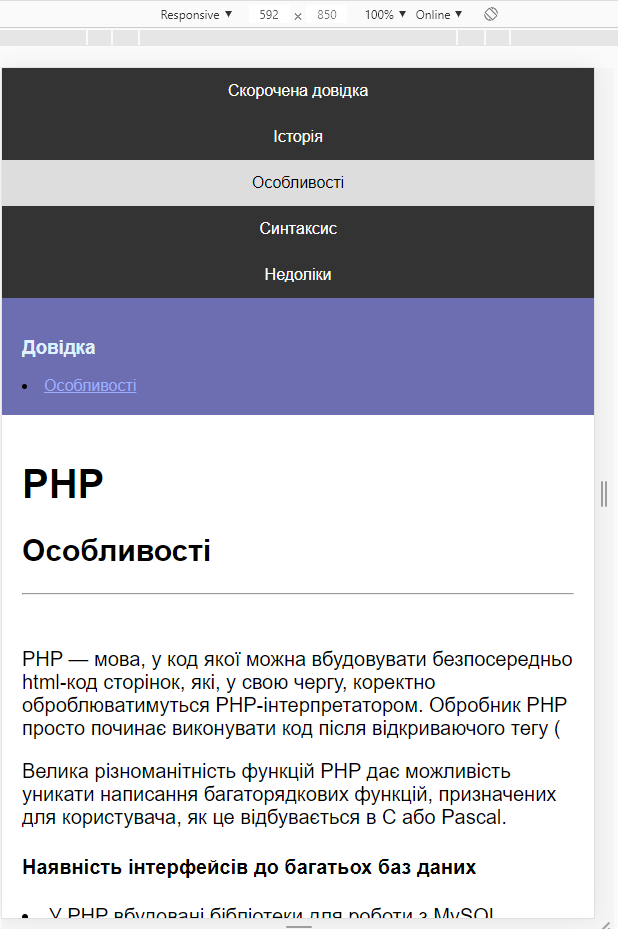
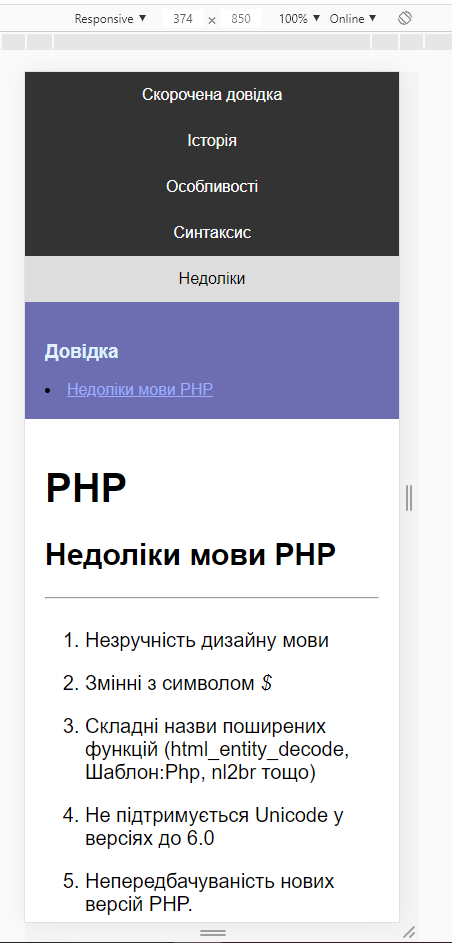
flex-direction: column;

}

}

Результат:



**Висновок**: Під час виконання лабораторної роботи я вивчив способи динамічного задання розмірів елементам сторінки в залежності від розміру пристрою для відображення. Я зіткнувся з проблемами відображення сторінки на малих розмірах екранів, та навчився усувати їх за допомогою css.

**Контрольні запитання:**

**Перерахуйте відомі вам web-браузери.**

Firefox, Safari, Google chrome, Opera, IE

**Що таке браузерні префікси в CSS властивостях?**

В деяких випадках CSS дозволяє задавати стилі окремо для різних браузерів за допомогою префіксів, кожен з яких є директивою відповідному браузеру

**Які браузерні префікси ви знаєте?**

префікс -moz- для Firefox, - ms- для Internet Explorer, -webkit- для Chrome і Safari, -o- для Opera.

**Для чого потрібні браузерні префікси в CSS властивостях?**

Їх використання виправдане, якщо необхідно забезпечити підтримку старих версій браузерів, а також для властивостей, що ще чітко не закріплені в специфікаціях та трактуються різними браузерами по-різному.

**Які CSS властивості можуть використовуватись з префіксами?**

display, transition, user-select, background тощо. (Загалом, для більшості)

**Що таке коментар з умовами?**

Спеціально для підтримки застарілих неоновлюваних версій браузера Internet Explorer 6-9 було впроваджено так звані коментарі з умовами (conditional comments). Контент, що розміщується між відкриваючим та закриваючим коментарями, буде застосовано тільки якщо версія браузера відповідає умові

**Які умови можна використовувати в коментарях з умовами?**

Приклад з умовою – версією браузера:

<!--[if IE 6]> Тільки для IE версії 6 <![endif]--> <!--[if IE 7]> Тільки для IE версії 7 <![endif]-->

**Яка область застосування коментарів з умовами?**

Підтримка застарілих версій браузерів

**Що таке роздільна здатність екрана?**

Число пікселів, яке може бути відображене екраном

**Для чого використовують мета-тег viewport?**

Щоб забезпечити адекватне відображення в першу чергу на мобільних пристроях, слід вказати браузеру, на яку ширину вікна розраховувати при побудові сторінки. Для цього використовують мета-тег viewport.

**Які атрибути може мати мета-тег viewport?**

Він може містити числове значення ширини в пікселях або у вигляді змінної device-width, початкове масштабування сторінки та обмеження для масштабування користувачем.

**Які одиниці виміру розмірів web-елементів ви знаєте?**

пікселі, відсотки, відносні одиниці(em), відношення до viewport

**Які переваги та недоліки побудови інтерфейсу із завданням розмірів у пікселях?**

Одна з проблем використання пікселів полягає в тому, що ці одиниці не дозволяють динамічно змінювати масштаб для мобільних пристроїв.

**Які переваги та недоліки побудови інтерфейсу із завданням розмірів у відносних одиницях?**

Перевагою використання відносних одиниць є динамічність.

**Яка різниця між одиницями виміру em та процентами?**

При установці властивості margin-left в %, відсоток береться від ширини батьківського блоку, а не від його margin-left. При установці властивості line-height в %, відсоток береться від поточного розміру шрифту, а не від line-height батька. Для width / height зазвичай відсоток від ширини / висоти батька, але при position: fixed, відсоток береться від ширини / висоти вікна (а не батька і не документа).

**Чим відрізняються адаптивний та респонсивний дизайн?**

Адаптивний дизайн містить багато варіантів сторінки, в той час як респонсивний лише один, але з багатьма налаштуваннями

**Який селектор активно використовується в інтерфейсі, керованому мишею, але не може бути використаний у інтерфейсі з сенсорним вводом?**

Hover

**Що таке медіа-запит?**

Правило @media дозволяє вказати тип носія, для якого буде застосовано вказаний стиль. В якості типів виступають різні пристрої, наприклад, принтер, монітор та ін.

**Які є способи завдання стилів за допомогою медіазапитів?**

Медіа запит можна використовувати як підключення до документа різних зв’язаних таблиць стилів, так і для виділення окремих глобальних стилів або як конструкцію у CSS файлі

**Які типи можна задавати в медіа-запиті?**

Як типи виступають різні пристрої, наприклад, принтер, монітор та ін. Найбільш відомі медіа-типи: screen, print, all.

**На що впливає тип, вказаний в медіа-запиті?**

Він визначає, для яких пристроїв буде застосовуватись правило.

**Які умови дозволяють встановлювати медіа-запити?**

max-width, max-height, min-width, min-height

**Які існують можливості комбінувати запити?**

Дозволяється використання операторів and та or.