

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Програмне рішення Інтернет-магазину для
впровадження на підприємстві»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Бабій М.С.

Студента групи ІІз – 61С

Кременчуков О.В.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Центр заочної, дистанційної і вечірньої форм навчання
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 г.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІНз-61С спеціальності “Інформатика”
заочної форми навчання Кременчукова Олексія Віталійовича.

Тема: “Програмне рішення Інтернет-магазину для впровадження на підприємстві”

Затверджена наказом по СумДУ

№ _____ від _____ 2020 г.

Зміст пояснювальної записки: 1) підходи до створення програмного продукту Інтернет-магазину; 2) постановка завдання; 3) мова програмування HTML; 4) основи CSS; 5) визначення мови програмування PHP; 6) опис мови JavaScript; 7) система управління базами даних MySQL; 8) архітектура web-додатку; 9) база даних та її адміністрування; 10) сценарії та основні модулі інформаційної системи; 11) відображення результатів роботи Web-застосунку.

Дата видачі завдання “ _____ ” _____ 2020 г.

Керівник випускної роботи _____ Бабій М.С.

Завдання прийняв до виконання _____ Кременчуков О.В.

РЕФЕРАТ

Записка: 74 стор., 18 рис., 16 додатків, 22 джерела.

Об'єкт дослідження — створення інтернет-магазину.

Мета роботи — розробити інформаційну систему в вигляді серверного веб-додатку для впровадження на підприємстві.

Методи дослідження — аналіз існуючих рішень, моделювання роботи Інтернет-магазину, тестування розробленої системи.

Результати — створено інформаційну систему (веб-застосунок) інтернет-магазин для впровадження на підприємстві.

Ключові слова:

ВЕБ-ДОДАТОК, СЕРВЕРНЕ ПРОГРАМУВАННЯ, ПРОГРАМУВАННЯ
МОВОЮ HTML, ПРОГРАМУВАННЯ МОВОЮ PHP, ПРОГРАМУВАННЯ
МОВОЮ JAVASCRIPT, ТАБЛИЦЯ СТИЛІВ CSS, ІНФОРМАЦІЙНА
СИСТЕМА, БАЗИ ДАНИХ MYSQL, ІНТЕРНЕТ-МАГАЗИН

ЗМІСТ

РЕФЕРАТ	3
ЗМІСТ	4
ВСТУП.....	6
1 ЛІТЕРАТУРНИЙ ОГЛЯД.....	8
1.1 Підходи до створення програмного продукту інтернет-магазину	8
1.2 Постановка завдання	10
2 ТЕОРЕТИЧНІ ОСНОВИ.....	11
2.1 Мова програмування HTML	11
2.2 Основи CSS	19
2.3 Визначення мови програмування PHP	20
2.4 Опис мови JavaScript	34
2.5 Система управління базами даних MySQL.....	34
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	38
3.1 Архітектура web-додатку	38
3.2 База даних та її адміністрування	39
3.3 Сценарії та основні модулі інформаційної системи	43
3.4 Відображення результатів роботи Web-застосунку	45
ВИСНОВКИ	47
СПИСОК ЛІТЕРАТУРИ.....	48
ДОДАТОК А	50
ДОДАТОК Б	55
ДОДАТОК В.....	56
ДОДАТОК Г	57

ДОДАТОК Г.....	58
ДОДАТОК Д.....	59
ДОДАТОК Е.....	64
ДОДАТОК Є.....	65
ДОДАТОК Ж.....	65
ДОДАТОК З.....	65
ДОДАТОК И.....	65
ДОДАТОК І.....	66
ДОДАТОК Ї.....	67
ДОДАТОК К.....	69
ДОДАТОК Л.....	71

ВСТУП

Однією з найважливіших завдань розробки веб-дизайну ресурсу є реалізація такої функції, як комунікативність. Інакше кажучи, дизайн сайту повинен бути такий, щоб користувач, опинившись на ньому, не просто закрив вкладку, але і виявив бажання більш детально ознайомитись зі змістом контенту. Не зважаючи на привабливий дизайн сучасних сайтів, їх основним недоліком є беззмістовність: відвідувач не може отримати потрібну інформацію відразу – йому потрібно здійснити безліч переходів між сторінками сайту, щоб досягти поставленої мети.

Довге успішне існування сайту можливе лише при гармоніюванні контенту та дизайну сайту. Розробка дизайну сайту – це створення так званого «мосту» для користувача і його метою є виключно шлях для досягнення цілі. Простий і наочний доступ до потрібної інформації – це мета якісного веб-дизайну, де, звичайно, можливе використання яскравих кольорів, тільки за умови, якщо це не викликає перевантаження і не впливає на зміст.

Веб-дизайн навігаційної системи ресурсу покликаний не тільки зорієнтувати відвідувача, де саме він знаходиться в даний момент, але й покликаний показати всю структуру. Сторінки повинні не тільки відображати інформацію, але й доносити до користувача повну ієрархію сайту, кожне посилання повинно бути вдало підібраним та розташованим. Оскільки сайт повинен бути присвячений якійсь тематиці, то всі посилання та зображення має вказувати саме на цю тему, якій присвячений сайт. Особливо детально варто підійти до наповнення, контенту сайту, тобто статті й новини, які він відображає мають відповідати тематиці, бути зрозумілими для відвідувачів. Важливою складовою є відображення контактної інформації, що підвищують довіру користувача. Також велике значення має вибір шрифтів: розміру, типу та кольору. Так, занадто дрібний або складний для сприйняття текст можуть ускладнити його прочитання. З іншого боку, текст назви компанії, розміщений і центрі сторінки та написаний великими літерами може швидко втомити

користувача, так само, як і, наприклад, недоречно розміщений логотип. Фірмові знаки, які обов'язково повинні бути присутні на кожному сайті, краще розміщувати в верхній частині сайту та обирати для їх оформлення ненав'язливий стиль. В залежності від бажання господаря сайту, розробник може виділяти або, навпаки приховувати певні елементи сайту для досягнення певних цілей та правильного розподілення уваги користувача. Для всіх типів веб-дизайну, одним з основних етапів є графічне зображення елементів сторінки.

Веб-дизайн сайту може візуально зацікавити користувача тільки тоді, коли гармонічно поєднується простота сприйняття та ненав'язлива естетичність.

Перед створенням сайту обов'язково слід дослідити та проаналізувати інші сайти, які відповідають обраній тематиці. Також варто звернути увагу на запити користувачів та розглянути сайти, які мають високу відвідуваність, ще, обов'язковим етапом дослідження є сайти з низьким рівнем відвідуваності, котрі потрібно відвідати, щоб не допустити подібних помилок. Раніше одним з основних показників розробки якісного сайту був швидкість завантаження сторінки, але, з розвитком технологій, ця характеристика має все менше значення. Наразі, коли обмеження по контенту майже зняті, доцільно використовувати flash-презентації, динамічні зображення, відеоролики та об'ємні тексти та графічні елементи.

Через те, що комп'ютери внутрішніх мереж багатьох підприємств мають низькі технічні характеристики, при розробці внутрішнього сайту підприємства часто доводиться відмовитися від багатьох можливостей веб-дизайну. Але в глобальній мережі Інтернет недостатня візуальна привабливість сайту часто приводить до втрати числа користувачів, чого не може дозволити собі жодна компанія, яка цінує увагу і інтерес відвідувачів.

1 ЛІТЕРАТУРНИЙ ОГЛЯД

1.1 Підходи до створення програмного продукту інтернет-магазину

Для успішного функціонування інтернет-магазину та збільшення доходу інтернет-магазину необхідно підібрати систему управління, яка б відповідала таким критеріям:

- можливість швидкої реєстрації та оформлення замовлення;
- застосування промо-кодів та автоматична система розрахунку знижок;
- зручний пошук по асортименту;
- простий і легкий редактор;
- можливість підключення та зручної інтеграції бухгалтерії підприємства;
- фільтр асортименту;
- можливість резервування певного товару;
- модуль інформування клієнтів електронною поштою.

1С Бітрікс – одна з найпопулярніших CMS для інтернет-магазину, яка зарекомендувала себе як система, що відповідає майже всім запитам користувачів. проте, необхідно звернути увагу на те, що Бітрікс надзвичайно витратна, тому частіше використовується тільки найбільшими інтернет-магазинами, але в цьому випадку адміністратор отримує безліч зручних можливостей і розширений функціонал.

Ця система відмінно синхронізується з 1С, що спрощує роботу бухгалтерії та оптимізує процес продажу товару через інтернет. Бітрікс відзначається високим ступенем захисту, чудово інтегрується з Яндекс. Маркет та Froogle, в системі можливо розподіляти права доступу для менеджерів та інтегрувати відразу декілька сайтів.

CS-Cart – система керування контентом сайту, застосувавши яку, можливо створити якісний інтернет-магазин. Веб-розробники запевняють, що ця CMS досить проста і функціональна та забезпечує багато корисних можливостей, серед яких варто відзначити:

- просте та зручне редагування асортименту та окремих товарів;

- автоматична оптимізація сайту для пошукових систем;
- можливість створення інформативних звітів про продажі;
- розрахунок вартості відзначається особливою гнучкістю;
- більше десяти платіжних систем для зручності покупців;
- мобільна версія сайту та адаптивний дизайн;
- синхронізація з 1С.

UMI.CMS – також користується популярністю серед розробників інтернет-магазинів. Ця система дуже малозатратна та пропонує непогані функціональні можливості для створення сайтів. Якщо розглядати переваги, то до них можна віднести інтеграцію з «Мой склад» та програмою 1С: Підприємство; мультимовність; великий вибір шаблонів для інтернет-магазинів; широкий спектр можливостей по створенню різноманітних звітів по замовленням покупців, середньому чеку покупок, кількості відвідувачів та багато іншого; можливість автоматичного оновлення.

Головна перевага даної системи – можливість створення магазину тільки за звичайною ціною віртуального хостинга.

Якщо говорити про створення інтернет-магазину на базі CMS, то потрібно обов'язково згадати ще й *Joomla*. Початкова мета цієї системи – створення корпоративних сайтів та розробки веб-візиток, але зараз вже доступні безліч модулів, що допомагають створити високо-функціональний інтернет-магазин. Для роботи даної системи обов'язкова наявність *VirtueMart*, без даної програми неможливе використання *Joomla*. Беззаперечною перевагою цієї системи є її швидкість, але окрім цього присутні й інші плюси:

- засоби додавання до товарів нових характеристик;
- ефективне управління групами клієнтів;
- налаштування цін по категоріям;
- управління сервісами логістики;
- формування звітності з заданими параметрами;
- інтеграція з PayPal, банківськими картами.

Shop-Script - CMS з найбільш широкими аналітичними можливостями. В цьому плані доступні інструменти створення когортного аналізу; організації спеціального спліт-тестування; різні види метрик; перегляд доходів за різними джерелами; створення звітів, прогнозів. Тому якщо веб-аналітика істотна, дана система управління стане хорошим вибором.

Amiro.CMS має досить багато інструментів для створення інтернет-магазину. Серед яких два плагіна: Вітрина і Бізнес. Вітрина - для створення електронного каталогу товарів, який має розширені опції для перегляду, порівняння, але не підходить для сайтів, яким необхідна функція продажів;

Бізнес - має всі можливості cms для великого інтернет-магазину, в тому числі широку систему рекламного, інформаційного та маркетингового забезпечення порталу.

OpenCart має корисні переваги – швидке завантаження даних. Для веб-розробника надається велика кількість шаблонів, серед яких можна вибрати гідний варіант для інтернет-магазину. За функціональними можливостями варто особливу увагу звернути на: детальну систему звітів, хороші функції для SEO оптимізації, можливість створювати знижки та акції, наявність партнерських програм, інтеграція з декількома платіжними системами.

1.2 Постановка завдання

Задача роботи – створення програмного рішення інтернет-магазину, розроблення інформаційного та програмного забезпечення інформаційної системи на основі використання баз даних та web-технологій, ознайомлення з сучасними технологіями передачі та обробки інформаційних потоків через Інтернет, архітектурою web-орієнтованих інформаційних систем, новітніми методами розроблення динамічних web-сайтів.

2 ТЕОРЕТИЧНІ ОСНОВИ

2.1 Мова програмування HTML

Мова розмітки Hypertext (HTML) - текстова мова для структурування електронних документів, таких як гіперпосилання, фотографії та ін. HTML-документи є основою Всесвітньої павутини та відображаються за допомогою веб-браузерів. На додаток до вмісту, що відображається браузером, HTML-файли можуть містити додаткову інформацію у вигляді метайнформації, наприклад, мови, які використовуються в тексті автора або узагальнений контент тексту.

HTML додатково розроблений Всесвітнім консорціумом Web (W3C) і Робочою групою з технологій застосування веб-гіпертекстів (WHATWG). Поточна версія HTML 5.2, з 14 грудня 2017 року, яка вже підтримується багатьма з веб-браузерів. Розширювана мова hypertext Markup Language (XHTML) також замінюється HTML5.

HTML використовується в якості мови для структурування тексту семантично, але не для його форматування. Візуальне уявлення не є частиною специфікацій HTML і визначається веб-браузером і шаблонами дизайну, такими як CSS. [21]

2.1.1 Синтаксис

Тексту надається структура за розміткою частин цього тексту.

Атрибут присвоюється за допомогою стандартизованих елементів. Більшість цих елементів HTML позначені парою тегів, тобто початковий тег і кінцевий тег. Тег початку завжди починається зі знака <. Далі йде назва елемента (наприклад, p для абзацу або h1 для заголовка першого порядку) та необов'язково список його атрибутів (наприклад, class = " warning" або id = " warning"). Тег початку закривається >. Кінцевий тег складається із символів </назви елемента>. Пов'язані теги початку та кінця разом із вмістом між ними

утворюють елемент загальної специфікації. Ці елементи можуть бути вкладені відповідно до правил, вказаних у визначенні типу документа:

`<p>` Абзац тексту, який містить слово `` підкреслене `</p>`

Певні атрибути не повинні бути чітко зазначені. Для деяких з них кінцевий тег може бути відсутній (наприклад, `</p>` або ``). Крім того, велика література не важлива для атрибутів (наприклад, ``, ``, ``). Для порівняння: в XHTML ці правила суворіші. [4]

Окрім атрибутів із початковими та кінцевими тегами, у HTML також є порожні атрибути, такі як розриви рядків (`br`) або зображення (`img`).

`
`Рядок тексту, який продовжується тут.

``

HTML - це описова процедура, а не процесуальна та орієнтована на презентацію розмітка тексту, навіть якщо HTML можна використовувати в попередніх версіях. Атрибути HTML - це інформація, яка вказує веб-браузеру, як візуально форматувати текст. Атрибут - це структурна одиниця, за допомогою якої певним текстовим областям можна присвоїти значення, наприклад, в `<h1>... </h1>` для заголовка, `<p>... </p>` для абзацу тексту та `... ` для підкресленого тексту. Яке значення в кінцевому підсумку передається користувачеві (у випадку заголовка, наприклад, збільшеним, жирним шрифтом) спочатку подається веб-браузеру та залежить від вихідного середовища. Оскільки HTML-документи зазвичай відображаються на екранах комп'ютерів, вони також можуть виводитися на інші носії інформації, наприклад, на папері або за допомогою голосового виводу. Стили CSS підходять для впливу на подання HTML-документа на різних носіях.

Читання вихідного коду і обробка існуючої інформації також згадується на технічній мові як розбір, а підготовка до висновку середовища, як рендеринг. Мова HTML описує, як браузер (або будь-яка інша програма, наприклад текстовий редактор) повинна «розуміти» текст, а не те, як він потім реалізує їх у візуалізації. Наприклад, в ньому йдеться про те, який заголовок слід використовувати, але не в якому розмірі шрифту або стилі він повинен бути

представлений - тільки деякі звичайні стандартні параметри стали натуралізованими, але вони не є частиною специфікації HTML. [17]

2.1.2 подача символів

Стандартний набір символів, спочатку створений на 7-бітному ASCII, був продовжений численними спеціальними символами, які закодовані в якості сутності HTML.

Для підтримки всіх мов світу потрібно підключення кодування UTF (Unicode), яке зараз реалізоване у всіх популярних браузерах. Таким чином, HTML призначений для незалежної від платформи переносимості, за умови, що вона підтримує HTML-рендерер. Базова подача символів для веб-документа обирається в мета-елементах, заголовку файлу, а потім браузер підлаштовується до нього.

Творці веб-сторінок, клавіатура яких не може надати всі символи безпосередньо, наприклад, німецькі умлаути, може кодувати спеціальні символи декількома способами; ("ä") може бути закодований як сутність HTML (`ä`), як десяткове значення Unicode (`ä`) або в кодуванні Юнікод (`&#U000000e4;`). Багато складних редакторів веб-сайтів автоматично вирішують спосіб кодування символів під час його здійснення.

URL-адресу символів, що безпосередньо не підтримуються, кодують за допомогою символів ASCII методом MIME. [19]

2.1.3 Тип мови

Спільною особливістю більшості мов програмування є те, що необхідно використовувати спеціальне програмне забезпечення для редагування вихідних документів, але, у випадку з HTML, будь-який текстовий редактор є достатнім.

2.1.4 Версії

HTML був вперше запропонований Тімом Бернерс-Лі в Женеві 13 березня 1989.

HTML (без номера версії, 3 листопада 1992): оригінальна версія на основі тільки тексту.

HTML (без номера версії, 30 квітня, 1993): на додаток до атрибутів, таких як жирний або курсивний текст, інтеграція зображень була додана до тексту.

HTML + (листопад 1993) заплановані розширення, які були включені в більш пізні версії, але не ПРИЙНЯТІ як HTML +.

HTML 2,0 (листопад 1995): версія, визначена в RFC 1866 введена технологія форм, серед іншого. Статус цього стандарту є історичним.

HTML 3,0: не випущено, оскільки ця версія застаріла з введенням браузера Netscape у версії 3 до запланованого випуску.

HTML 3,2 (14 січня 1997): нові функції, такі як таблиці, текстові потоки навколо зображень, інтеграція аплетів.

HTML 4,0 (18 грудня 1997): введення стилів, скрипти, кадри. Трохи виправлена версія була випущена 24 квітня 1998.

Html 4,01 (24 грудня 1999): замінено HTML 4,0 з багатьма незначними виправленнями. Була стандартною протягом тривалого часу до 2014.

XHTML 1,0 (26 січня 2000): переформулюванні HTML 4,01 за допомогою XML. Переглянута версія була випущена 1 серпня 2002 року.

Xhtml 1,1 (31 травня 2001): після XHTML була розділена на МОДУЛІ, XHTML 1,1 визначив сувору версію, яка усунула набір рамок і перехідних варіантів з HTML 4.

XHTML 2,0 (закритий, 26 липня 2006): Ця версія більше не використовувала HTML 4,01 і ввела деякі нові елементи, такі як для списки навігації. W3C перестала працювати на XHTML 2,0 влітку 2009, тому що XHTML був замінений на HTML5.

HTML5 (28 жовтня 2014): створення нової лексики на основі HTML 4,01 та XHTML 1,0. Специфікації DOM, пов'язаної з HTML, також були переглянуті та розширені.

HTML 5,1 (1 листопада 2016)

HTML 5,2 (14 грудня 2017): остання версія. [20]

2.1.5 Структура

Загальна структура

HTML-документ складається з трьох областей:

тип документа декларації (doctype) на самому початку файлу, який визначає тип документа (DTD), наприклад, HTML 5,

HTML-заголовок (`<head>`), який містить в основному технічну або документальну інформацію, яка зазвичай не відображається в області відображення браузера HEAD

HTML Body (`<body>`) - містить інформацію, яка зазвичай відображається в області відображення браузера. BODY

Таким чином, базова структура веб-сторінки виглядає так:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Заголовок</title>
  </head>
  <body>
    <p>Текст сайту... </p>
  </body>
</html>
```

Заголовок HTML

В основному можна використовувати сім різних елементів:

`title` - позначає назву сторінки, яка відображається в рядку заголовка у більшості браузерів.

`meta` - може містити різні метадані.

`base` - вказує базовий URI.

link - використовується для зазначення логічних зв'язків з іншими ресурсами. Найбільш часто використовується для інтеграції таблиць стилів.

script - інтегрує код в конкретну скриптову мову, в основному JavaScript.

style - містить інформацію про стиль, в основному декларації CSS.

object - прив'язує зовнішній файл. Браузери не повинні відображати такі об'єкти у заголовку документа. Починаючи з HTML5, тег об'єкта більше не дозволений у заголовку HTML. [3]

Тіло HTML

Тіло HTML містить фактичну інформацію про сторінку. HTML відрізняє між блоком і вбудованими елементами. Основна відмінність полягає в тому, що перші в виході створюють власний блок, в якому розміщується вміст, в той час як вбудовані елементи не переривають потік тексту. Простіше кажучи, блокові елементи завжди мають свій власний абзац. Однак, CSS дає можливість представляти блокові елементи як вбудований елемент, і навпаки. Крім того, всі елементи також можуть бути позначені як вбудований блок за допомогою CSS, з результатом чого такий елемент має як властивості блоку елемента, так і вбудований елемент.

Заголовок першого порядку представлений такими тегами:

```
<h1>Заголовок 1</h1>
```

h1 позначає заголовок 1, який є заголовком першого (і найвищого) рівня структури в HTML. Також можливо використовувати заголовки від другого до шостого рівня (h2-h6) [4]

Гіперпосилання

```
<a href="http://example.com/">Це посилання на example.com</a>
```

Гіперпосилання є посиланнями на інші ресурси, як правило, також HTML документів, які зазвичай можуть бути записані в браузері та відкриваються натиснувши. Це посилання може бути візуалізоване таким чином: це посилання на example.com, аналогічно, цей приклад показує, що атрибут Link є вбудованим.

Звичайний текст вказаний за замовчуванням (для абзацу), його можна записувати без тегів, але для використання стилів CSS необхідно зазначити тег `<p></p>`, що позначає абзац.

Щоб вивести текст у форматі HTML:

`<p>` Це зразок тексту `</p>`

Для виділення текстів жирним шрифтом або курсивом використовуються теги `` та ``

Опис структури тексту дозволяє легко налаштувати візуалізацію для глядача, наприклад, читати текст до особи з вадами зору або виводити його як Шрифт Брайля. [15]

HTML-варіанти

При проектуванні останньої версії HTML 4, той факт, що багато документів HTML як і раніше використовувати елементи і атрибути для візуалізації. В результаті було розроблено три варіанти:

Строгий - цей DTD включає в себе основний набір елементів і атрибутів. Більшість елементів і атрибутів відсутні, щоб впливати на візуалізацію, включаючи елементи та атрибути, такі як, і. Текстові та неблочно утворюючі елементи і повинні бути завжди в контейнері елементом, наприклад, членом а.

Перехідний - варіант переходу містить навіть старі елементи та атрибути, які також дозволяють робити розмітку фізичних текстів. Мета цього DTD полягає в тому, щоб веб-автори, які ще не відокремлюють логічне структурування та візуалізацію для запису HTML-сумісних стандартів. У той же час вона призначена для забезпечення того, щоб існуючі веб-сторінки все ще могли розглядатися через поточні веб-браузери.

Набір рамок - крім всіх елементів перехідного варіанту, цей варіант також містить елементи для створення рамок. [18]

2.1.6 Додаткові прийоми та події

Каскадні таблиці стилів

Протягом багатьох років, HTML був розширений, щоб включити елементи, які служать для візуального проектування документів. Це суперечить оригінальній ідеї системної незалежності. Повернення до поділу структури і макета було зроблено шляхом визначення каскадних таблиць стилів (CSS). Дизайн або зовнішній вигляд документа в окремому файлі має бути визначений так званим списком стилів. Це покращує адаптивність макета до відповідного пристрою виводу і до конкретних потреб користувачів, наприклад, спеціальний дисплей для людей з вадами зору. В даний час, підтримка CSS браузерів є достатньою для реалізації складного дизайну. [5]

CSS-файл може бути включений в заголовок HTML за допомогою тега Link:

```
<link rel="stylesheet" href="stylesheet.css">
```

Динамічний HTML

Дуже рано в історії HTML, були винайдені додаткові методи, які дозволяють динамічно змінювати HTML документи під час відображення в браузері. Найпоширенішим є JavaScript. У таких інтерактивних документах згадується динамічний HTML. Ці методи були розроблені незалежно від різних виробників браузерів, особливо Microsoft і Netscape. Таким чином, були істотні проблеми у впровадженні методів між різними браузерами. Тим часом, всі JavaScript-браузери з підтримкою інтерпретації документа об'єктної моделі (DOM), що дає змогу писати сценарії, які можна запустити у всіх браузерах. Тим не менш, є ще відмінності в підтримці стандарту DOM.

XHTML

XHTML 1,0 був розроблений на базі HTML 4,01 (SGML). XHTML задовольняє більш суворі синтаксичні правила XML в порівнянні з SGML, але семантично ідентичні відповідному DTD варіанту HTML 4,01 в трьох його варіантах DTD.

HTML5

Відповідні переваги SGML і XML попередніх версій HTML були об'єднані в HTML5. На відміну від попередніх версій HTML, у HTML5 більше немає DTD.

AJAX

З технологією AJAX, JavaScript дозволяє конкретно змінити і перезавантажити окремих вміст веб-браузера, який вже був завантажений без необхідності перезавантажувати сайт повністю. З одного боку, нижній обсяг даних дозволяє забезпечити швидку відповідь веб-сервера, з іншого боку, можна імітувати спосіб настільних додатків реагувати. [17]

2.2 Основи CSS

CSS був розроблений таким чином, щоб значною мірою відокремити специфікації відображення від змісту. Якщо це розділення здійснюється послідовно, описується лише структура вмісту документа та значення його частин у HTML або XML, тоді як з CSS окремо, переважно в окремих CSS-файлах, визначається подання вмісту (наприклад, макет, кольори). На початку були лише прості інструкції із відображенням, з часом додавались більш складні модулі, за допомогою яких, наприклад, анімації та різні зображення можуть бути визначені для різних вихідних носіїв.

Елементи документа можна ідентифікувати на основі різних властивостей. На додаток до імені елемента (наприклад, для всіх гіперпосилань), їх ідентифікатора та їхнього положення в документі (наприклад, всі елементи зображення в елементах посилання), це також включає такі деталі, як атрибути (наприклад, усі елементи посилання, href-атрибут починається з `www.example.com`) або положення в наборі елементів (наприклад, кожен непарний елемент у списку). Інструкції CSS можна використовувати для визначення специфікацій для відображення кожної такої групи елементів. Ці характеристики можливо скласти централізовано, навіть в окремих файлах, щоб їх можна було легко використовувати для інших

документів. CSS також включає модель успадкування розмітки, яка зменшує кількість необхідних визначень.

За допомогою CSS можна вказати різні подання для різних вихідних носіїв (екран, папір, проекція, мова). Це корисно для, наприклад, перелічення довідкових адрес гіперпосилань під час друку та адаптивність для таких пристроїв, як КПК та мобільні телефони, які мають невеликі дисплеї або низьку роздільну здатність, зображення, які є досить вузькими та не надто високими, щоб залишатись розбірливими на таких пристроях.

CSS - це стандартна мова стилів у всесвітній мережі. Раніше поширені HTML-елементи, орієнтовані на візуалізацію, такі як шрифт, вважаються застарілими, а це означає, що їх потрібно буде видалити зі стандарту HTML у майбутньому, наприклад, оскільки HTML 4 (1997 р.) елементи цієї мови вважаються "небажаними" та застарілими в HTML5. [5]

2.3 Визначення мови програмування PHP

PHP – одна з найпопулярніших мов програмування, яка, в основному, призначена для веб-програмування. Вона розроблена датським і канадським програмістом Рамумусом Лердорфом в 1994.

PHP-код, як правило, обробляється на веб-сервері інтерпретатором PHP здійснюється як модуль або як загальний шлюз інтерфейсу, виконуваний файл. На веб-сервері інтерпретований і виконаний PHP-код, який може нести в собі будь-який тип даних, наприклад згенеровані HTML або двійкові дані зображення - буде формувати всю або частину відповіді http . Існують різні веб-шаблонні системи, веб- системи управління контентом, і веб-фреймворки, які можуть бути використані для оркекат або сприяти генерації цієї відповіді. Крім того, PHP може бути використаний для багатьох завдань програмування за межами веб-контексту, таких як автономні графічні додатки та програми управління безпілотника . Довільний PHP-код також може бути

інтерпретований і виконаний за допомогою інтерфейсу командного рядка (CLI). [1]

Стандартний інтерпретатор PHP, що працює на Zend Engine, є вільним програмним забезпеченням, випущеного під ліцензією PHP. PHP був широко портований і може бути розгорнутий на більшості веб-серверів практично на кожній операційній системі й платформі безкоштовно.

Мова PHP розвивалися без офіційних письмових специфікацій (стандартів) до 2014, з оригінальною реалізацією, що діє як стандарт де-факто які інші реалізації спрямовані на наслідування. З 2014, робота пройшла на створення офіційної специфікації PHP.

Станом на квітень 2020 року, більше половини сайтів в Інтернеті створено за допомогою PHP. З версії 7,0 і 7,1 більше 68%, які офіційно не підтримуються командою розробників PHP, в той час як підтримка безпеки забезпечується третіми сторонами, ТАКИМИ як Debian (до червня 2020 для PHP 5). Через популярність PHP (принаймні 53% веб-сайтів у світі працюють на реалізації мови) більше не підтримуються їхніми дизайнерами. Крім того, PHP версії 7,2 (найпопулярніша підтримувана версія PHP) - буде припинити отримувати оновлення безпеки 30 листопада 2020. [12]

2.3.1 Синтаксис

У наступному "Привіт, світ!" PHP-код вбудований в HTML документ:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP "Привіт, світ!" program</title>
  </head>
  <body>
    <?php echo '<p>Hello, World!</p>'; ?>
  </body>
</html>
```

Однак, не існує вимог для PHP-коду, що вбудований в HTML, найпростіший варіант Привіт, світ! може бути таким чином, без тега закриття, що краще в файлах, які містять чистий код PHP.

```
<?='Привіт, світ!';
```

Компілятор тільки виконую PHP-код, який записаний між роздільниками. Все що не входить до роздільників не обробляється компілятором, хоча, ко, який не виділений, як PHP все ще підлягає контролю структур, описаних в кодї. Найчастіше використовується форма запису, з використанням роздільників відкриття та закриття. Також існує скорочена форма запису, яка не містить тегів закриття. Короткий запис зустрічається рідше, адже вона робить файли сценарію менш портативними, оскільки підтримка для них часто відключена в конфігурації PHP. До PHP 5.4.0, цей короткий синтаксис працює тільки з параметром конфігурації включень, а для PHP 5.4.0 і пізніше він завжди доступний. Мета всіх цих обмежувачів полягає в окремому кодї PHP з не-PHP змісту, таких як JavaScript коду або HTML розмітки.

```
<?php?><?<?=echo()short_open_tag
```

Коротка форма запису актуальна в XHTML та інших XML документах та створює правильно сформовані інструкції з обробки XML. це означає, що результуюча суміш коду PHP та іншої розмітки в серверному файлі сама по собі є добре сформованою XML. `<?php?>`

Тип не потрібно вказувати заздалегідь, змінні починаються із значка долара. PHP 5 представив тип даних, який вказує на функції, щоб параметри конкретного класу – масиви, інтерфейси чи функції. Проте, до версії PHP 7, неможливо виразити типи даних, такі як ціле число або рядок.

Імена змінних залежать від регістру, на відміну від функцій. За допомогою подвійних лапок і heredoc можливо інтерпретувати значення змінної в рядок. Новий рядок в мові PHP виділяється Крапкою з комою. В даній мові програмування можна використовувати три типи коментарів: `/**/`, `//`, та `#echo`.

PHP, з погляду на ключові слова, найбільше подібна до мови програмування C, а якщо брати до уваги такі елементи як умови, цикли і повертає функції аналогічні за синтаксисом для мов, таких як C, C++, C#, Java і Perl. `ifforwhile`. [8]

2.3.2 Типи даних

PHP зберігає числа цілі та з плаваючою точкою, залежно від розрядності платформи формується бітність числа (32, 64 або 128), що подібно до типу даних мови C. Також, подібно до багатьох інших мов програмування, непідписані числа можуть бути перетворені в підписані значення. Цілі змінні задаються додатною та від'ємною десятковою, вісімковою, шіснадцятковою та двійковою системами числення.

Числа з плаваючою точкою можна вказувати за допомогою нотації з плаваючою точкою або двох форм наукового позначення. В PHP присутні типи даних, які притаманні мовам програмування Java та C++, а також власний логічний тип. За правилами перетворення логічних типів, нульові значення інтерпретуються як сутності типу false (так само, як на Perl й C++). NULL використовується як тип даних, який записується для інтерпретації порожніх значень.

Змінні типу "ресурс" відображають посилання на ресурси з зовнішніх джерел. Вони зазвичай створюються функціями з певного розширення, і можуть оброблятися лише функції одного розширення. Приклади включають файли, зображення та ресурси бази даних.

В масивах можуть міститися елементи будь-якого типу, що обробляється за допомогою PHP та може включати навіть інші масиви. Порядок зберігається у списках значень і в хеш-значеннях з обома ключами та значеннями. PHP також підтримує рядки, які можуть бути використані з одинарними лапками, подвійними лапками, `nowdoc` або `heredoc` синтаксисом.

Стандартна бібліотека PHP (SPL) намагається вирішити стандартні проблеми і реалізує ефективні інтерфейси доступу до даних і класи. [7]

2.3.3 Функції

PHP визначає великий масив функцій на основній мові, і багато з них також доступні в різних додатках; ці функції добре зафіксовані в онлайн-

документації PHP. Однак вбудована бібліотека має широкий спектр умов імен та пов'язані з ними невідповідності, як описано в історії вище.

Спеціальні функції можуть бути визначені розробником:

```
unction myAge(int $birthYear): string
{
    $yearsOld = date('Y') - $birthYear;
    return $yearsOld . ' year' . ($yearsOld != 1 ? 's':'');
}
echo Мені зараз ' . myAge(1995) . ';
```

У 2020 році результатом вищезгаданої вибіркової програми є «Мені зараз 25».

Замість покажчиків функцій на функції PHP можна посилатись рядком, що містить їх ім'я. Таким чином, звичайні функції PHP можуть використовуватися, наприклад, як зворотні дзвінки або в межах функціональних таблиць. Функції, визначені користувачем, можуть бути створені в будь-який час без прототипу. Функції можуть бути визначені всередині кодових блоків, що дозволяють вирішити час виконання того, чи слід визначати функцію. Існує функція `function_exists`, яка визначає, чи вже визначена функція із заданим іменем. Виклики функцій повинні використовувати дужки, за винятком функцій конструктора класу нульових аргументів, викликаних оператором PHP `new`, і в цьому випадку дужки є необов'язковими. [9]

До PHP 5.3, підтримка анонімних функцій не існувала в PHP. Хоча `create_function()` існує з PHP 4.0.1, це лише тонка обгортка навколо `eval()`, яка дозволяє створювати нормальні функції PHP під час виконання програми. PHP 5.3 додала синтаксис для визначення анонімної функції або "закриття", яка може захоплювати змінні з оточуючої області:

```
function getAdder($x)
{
    return function($y) use ($x) {
        return $x + $y;
    };
}
$adder = getAdder(8);
echo $adder(2); // результат "10"
```


У наведеному вище прикладі функція `getAdder ()` створює закриття, використовуючи переданий аргумент `$x` (ключове слово використовує імпорт змінної з лексичного контексту), який приймає додатковий аргумент `$y`, і повертає створене закриття абоненту. Така функція є об'єктом першого класу, тобто вона може зберігатися в змінній, передаватися як параметр іншим функціям тощо. [10]

Незвично для мови, що динамічно набирається, PHP підтримує декларації типу щодо параметрів функції, які застосовуються під час виконання. Це підтримується для класів та інтерфейсів з PHP 5.0, для масивів з PHP 5.1, для "callable" з PHP 5.4 та скалярних (цілочисельний, плаваючий, рядковий та булевий) типи з PHP 7.0. PHP 7.0 також має декларації типів для типів повернення функцій, виражених розміщенням імені типу після списку параметрів, яким передують двокрапки. Наприклад, функцію `getAdder` з попереднього прикладу можна зазначити такими типами як у PHP 7:

```
function getAdder(int $x): Closure
{
    return function(int $y) use ($x): int
    {
        return $x + $y;
    };
}
$adder = getAdder(8);
echo $adder(2); // друкує "10"
echo $adder(null); // викидає виняток, оскільки був переданий
неправильний тип
$adder = getAdder([]); // також кинув би виняток
```

За замовчуванням декларації скалярного типу відповідають слабким принципам введення тексту. Так, наприклад, якщо тип параметра `int`, PHP дозволить не тільки цілі числа, а й конвертовані числові рядки, поплавці чи булеві символи передавати цій функції і перетворив би їх. Однак PHP 7 має режим "суворого набору тексту", який при використанні вимикає такі перетворення для викликів функцій та повертається у файл. [13]

2.3.4 Створення та реалізація

Єдиною повною реалізацією PHP є оригінал, відомий просто як PHP. Він є найбільш широко використовуваним і працює від Zend Engine. Щоб відключити його від інших реалізацій, іноді неофіційно його називають "Zend PHP". Zend Engine автоматично збирає вихідний код PHP у внутрішній формат, який він може виконувати, таким чином він працює як інтерпретатор. Це також "опорна реалізація" PHP, оскільки PHP не має формальної специфікації, і тому семантика Zend PHP визначає семантику PHP. Завдяки складній та нюансованій семантиці PHP, визначеній як працює Zend, конкуруючим реалізаціям важко запропонувати повну сумісність.

Модель одного запиту на виконання сценарію PHP та факт, що Zend Engine є інтерпретатором, призводить до неефективності; в результаті були розроблені різні продукти, які допомагають поліпшити продуктивність PHP. Щоб пришвидшити час виконання і не доведеться компілювати вихідний код PHP щоразу, коли доступ до веб-сторінки, сценарії PHP також можна розгорнути у внутрішньому форматі двигуна PHP, використовуючи кеш-код опкоду, який працює за допомогою кешування складеної форми скрипт PHP (опкоди) у спільній пам'яті, щоб уникнути накладних розборів та компіляції коду кожного разу, коли сценарій запускається. Кеш опкодів Zend Opcode вбудований у PHP з версії 5.5. Іншим прикладом широко використовуваного кешу опкоду є альтернативний кеш PHP (APC), який доступний у вигляді розширення PECL. [11]

Хоча Zend PHP досі залишається найпопулярнішою реалізацією, було розроблено кілька інших реалізацій. Деякі з них є компіляторами або підтримують компіляцію JIT, а значить, пропонують переваги продуктивності над Zend PHP за рахунок відсутності повної сумісності PHP. Альтернативні реалізації включають наступне:

HHVM (HipHop Virtual Machine) - розроблений у Facebook і доступний як відкритий код, він перетворює PHP-код у байт-код високого рівня (зазвичай

відомий як проміжна мова), який потім динамічно переводиться на машинний код x86-64 під час виконання щойно вчасно (JIT) компілятор, в результаті чого до 6× поліпшення продуктивності. Однак, оскільки версія 7.2 Zend перевершила HHVM, і HHVM 3.24 є останньою версією, яка офіційно підтримує PHP.

Parrot - віртуальна машина, призначена для ефективного запуску динамічних мов; Pirc перетворює вихідний код PHP в проміжне представлення Parrot, яке потім переводиться в байт-код Parrot і виконується віртуальною машиною.

ReachPie - компілятор другого покоління для байтового коду .NET Common Intermediate Language (CIL), побудований на платформі Roslyn; наступник Фаланжера, що ділиться кількома архітектурними компонентами

Phalanger - компілює PHP у байт-код загальної проміжної мови (CIL); попередник ReachPie

Quercus - компілює PHP в байт-код Java

HipHop - розроблений у Facebook і доступний як відкритий код, він перетворює PHP-скрипти в код C++ і потім компілює отриманий код, зменшуючи завантаження сервера до 50%. На початку 2013 року Facebook знецінив це на користь HHVM через кілька причин, включаючи труднощі з розгортанням та відсутність підтримки всієї мови PHP, включаючи конструкції `create_function ()` та `eval ()`. [22]

2.3.5 Ліцензування

PHP - це безкоштовне програмне забезпечення, випущене під ліцензією PHP, яке передбачає, що:

Продукти, отримані з цього програмного забезпечення, не можуть називатися "PHP", а також "PHP" не можуть відображатися на їх назві без попереднього письмового дозволу group@php.net. Ви можете вказати, що ваше

програмне забезпечення працює спільно з PHP, кажучи "Foo for PHP", а не називати його "PHP Foo" або "phpfoo".

Це обмеження щодо використання "PHP" робить Ліцензію PHP несумісною з Загальнодоступною ліцензією (GPL), тоді як ліцензія Zend несумісна через рекламне застереження, подібне до оригінальної ліцензії BSD. [18]

2.3.6 Розвиток та співтовариство

PHP включає різні безкоштовні бібліотеки з відкритим кодом у своєму розповсюдженні джерел або використовує їх у отриманих бінарних збірках PHP. PHP - це по суті Інтернет-система з вбудованими модулями для доступу до серверів протоколу передачі файлів (FTP) та багатьох серверів баз даних, включаючи PostgreSQL, MySQL, Microsoft SQL Server та SQLite (що є вбудованою базою даних), серверами LDAP та ін. Численні функції, знайомі програмістам на C, наприклад, в сім'ї stdio, доступні в стандартних PHP-версіях.

PHP дозволяє розробникам писати розширення на C, щоб додати функціональність мові PHP. Розширення PHP можна складати статично в PHP або динамічно завантажувати під час виконання. Для розширення підтримки API Windows, управління процесами в Unix-подібних операційних системах, багатобайтових рядків (Unicode), cURL та декількох популярних форматів стиснення написано численні розширення. Інші функції PHP, доступні через розширення, включають інтеграцію з IRC, динамічне генерування зображень та вмісту Adobe Flash, об'єкти даних PHP (PDO) як рівень абстракції, що використовується для доступу до баз даних, і навіть синтез мови. Деякі основні функції мови, наприклад ті, що стосуються рядків і масивів, також реалізовані як розширення. Проект бібліотеки спільноти розширень PHP (PECL) є сховищем для розширень до мови PHP. [13]

Деякі інші проекти, наприклад, Zephir, надають можливість створювати розширення PHP мовою високого рівня та компілювати в рідні розширення

PHP. Такий підхід замість того, щоб писати розширення PHP безпосередньо на C, спрощує розробку розширень та скорочує час, необхідний для програмування та тестування.

Станом на грудень 2018 року до групи PHP входили десять осіб: Тіес Арнтцен, Стіг Баккен, Шейн Каравео, Енді Гутманс, Расмус Лердорф, Сем Рубі, Саша Шуман, Зеєв Сураскі, Джим Уінстед та Андрій Змієвський.

Zend Technologies надає сертифікацію PHP на основі іспиту PHP 5.5 для програмістів, щоб стати сертифікованими розробниками PHP. [11]

2.3.7 Використання

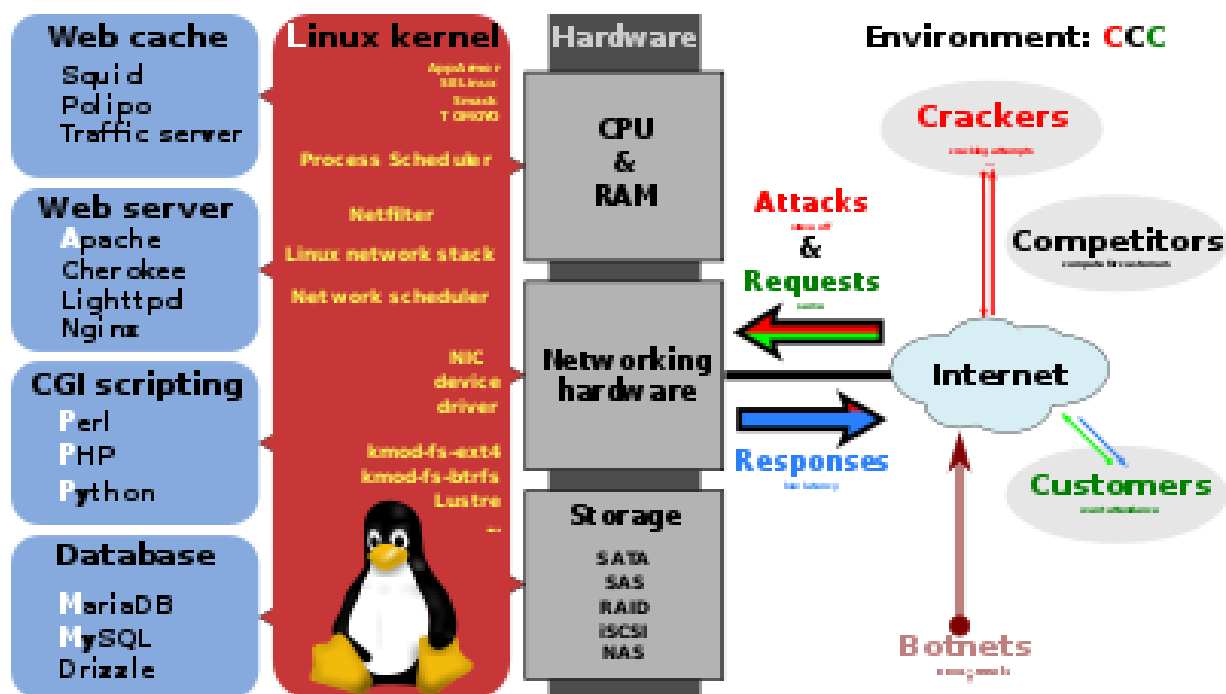


Рисунок 2.1 - Широкий огляд набору програмного забезпечення LAMP, відображеного тут разом із Squid

PHP - це сценарій мови загального призначення, який особливо підходить для веб-розробки веб-серверів, і в цьому випадку PHP зазвичай працює на веб-сервері. Будь-який код PHP у запитованому файлі виконується під час виконання PHP, як правило, для створення динамічного вмісту веб-сторінки або динамічних зображень, що використовуються на веб-сайтах чи в інших місцях.

Він також може використовуватися для сценаріїв командного рядка та додатків графічного користувальницького інтерфейсу (GUI) на базі клієнта. PHP може бути розгорнутий на більшості веб-серверів, багатьох операційних системах і платформах і може використовуватися з багатьма системами управління реляційними базами даних (RDBMS). Більшість постачальників веб-хостингів підтримують PHP для використання їх клієнтами. Він доступний безкоштовно, а група PHP надає повний вихідний код для побудови, налаштування та розширення для власного користування.



Рисунок 2.2 - Динамічна веб-сторінка: приклад сценаріїв на стороні сервера (PHP та MySQL)

PHP діє в основному як фільтр, приймаючи введення з файлу або потоку, що містить текстові та / або інструкції PHP, і виводить інший потік даних. Найчастіше вихідним буде HTML, хоча це можуть бути JSON, XML або двійкові дані, такі як формати зображення або аудіо. Починаючи з PHP 4, PHP-аналізатор збирає вхід для створення байтового коду для обробки Zend Engine, забезпечуючи покращену продуктивність у порівнянні з попередником інтерпретатора.

Спочатку призначений для створення динамічних веб-сторінок, PHP зараз зосереджується в основному на сценаріях на стороні сервера, і це аналогічно іншим мовам скриптування на стороні сервера, які надають динамічний контент від веб-сервера до клієнта, наприклад, Microsoft ASP.NET Microsoft, Сторінки JavaServer, Sun Microsystems, та mod_perl. PHP також залучила розробку багатьох програмних рамок, які надають будівельні блоки та структуру дизайну для сприяння швидкому розвитку додатків (RAD). Деякі з

них включають PRADO, CakePHP, Symfony, CodeIgniter, Laravel, Yii Framework, Phalcon та Zend Framework, пропонуючи функції, схожі на інші веб-рамки.

Архітектура LAMP стала популярною у веб-індустрії як спосіб розгортання веб-додатків. PHP зазвичай використовується як P у цьому пакеті поряд з Linux, Apache та MySQL, хоча P може також посилатися на Python, Perl або якусь комбінацію цих трьох. Подібні пакети, WAMP та MAMP, також доступні для Windows та macOS, причому перша буква відповідає відповідній операційній системі. Хоча і PHP, і Apache передбачені як частина встановлення бази macOS, користувачі цих пакетів шукають більш простий механізм встановлення, які можна простіше оновлювати. [6]

Станом на квітень 2007 року у понад 20 мільйонів Інтернет-доменів розміщено веб-сервіси, розміщені на серверах із встановленим PHP, і mod_php був зафіксований як найпопулярніший модуль сервера HTTP-сервера Apache. Станом на серпень 2019 р. PHP використовувався як серверна мова програмування на 78,8% веб-сайтів, що зменшилося від 83,5% раніше, де мова могла бути визначена. Системи управління веб-контентом, написані на PHP, включають MediaWiki, Joomla, eZ Publish, платформу eZ, SilverStripe, WordPress, Drupal, та Moodle. Веб-сайти, написані на PHP, у зворотній частині та / або націленій на користувачів частині, включають Facebook, Digg, Tumblr, Dailymotion, та Slack.

Для конкретних та більш досконалих сценаріїв використання PHP пропонує чітко визначений та задокументований спосіб написання призначених розширень на C або C++. Окрім розширення самої мови у вигляді додаткових бібліотек, розширення - це спосіб покращити швидкість виконання там, де вона є критичною, і є можливість для вдосконалень за допомогою справжньої компільованої мови. PHP також пропонує чітко визначені способи вбудовування себе в інші програмні проекти. Таким чином, PHP може бути легко використаний як внутрішня мова скриптів для іншого проекту,

забезпечуючи також тісне взаємодія з конкретними внутрішніми структурами даних проекту.

PHP отримав неоднозначні огляди через відсутність підтримки багатопотокового читання на рівні основної мови, хоча використання потоків стає можливим завдяки розширенню PECL "pthreads". [9]

Станом на січень 2013 р. PHP використовувався на понад 240 мільйонах веб-сайтів (39% із вибірки) та встановлювався на 2,1 мільйона веб-серверів.

Було створено інтерфейс командного рядка, php-cli та два двигуни сценаріїв хост-сценарію ActiveX Windows для PHP.

Станом на квітень 2020 р., PHP 7 є найбільш вживаною версією мови [220], де 50,6% сайтів використовують PHP та 39,6% усіх веб-сайтів у мережі Інтернет.

2.3.8 Безпека

У 2019 році 11% усіх уразливостей, перелічених у Національній базі даних про вразливість, були пов'язані з PHP; історично близько 30% усіх уразливостей, перелічених з 1996 року в цій базі даних, пов'язані з PHP. Недоліки в технічній безпеці самої мови або її основних бібліотек не є частими (22 у 2009 р. - близько 1% від загальної кількості, хоча PHP стосується приблизно 20% перелічених програм). Визнаючи, що програмісти роблять помилки, деякі мови включають перевірку зворотності, щоб автоматично виявити відсутність перевірки вводу, що викликає багато проблем. Така особливість розробляється для PHP, але її включення до випуску було відхилено кілька разів у минулому.

Існують вдосконалені патчі захисту, такі як Suhosin та Hardening-Patch, спеціально розроблені для середовищ веб-хостингу.

Історично старі версії PHP мали деякі параметри конфігурації та значення за замовчуванням для таких параметрів виконання, що робило деякі програми PHP схильними до проблем безпеки. Серед них, найвідомішими були

директиви конфігурації `magic_quotes_gpc` та `register_globals`; остання перетворила будь-які параметри URL-адреси на змінні PHP, відкривши шлях до серйозних уразливостей безпеки, дозволяючи зловмиснику встановити значення будь-якої неініціалізованої глобальної змінної та перешкоджати виконанню сценарію PHP. Підтримка налаштувань "магічних лапок" та "зареєструвати глобальні реєстри" застаріла з PHP 5.3.0 та видалена з PHP 5.4.0. [7]

Інший приклад можливої вразливості налаштувань часу виконання - це неможливість відключення виконання PHP (наприклад, за допомогою директиви конфігурації двигуна) для каталогу, де зберігаються завантажені файли; включення цього може призвести до зловмисного коду, вбудованого у завантажені файли. Найкраща практика - або знайти каталог зображень за межами кореня документа, доступного для веб-сервера, і обслуговувати його через посередницький скрипт, або відключити виконання PHP для каталогу, в якому зберігаються завантажені файли.

Таким чином, включення динамічного завантаження розширень PHP (через директиву `enable_dl` конфігурації) у спільному середовищі веб-хостингу може призвести до проблем із безпекою.

Перетворення типових типів, які призводять до того, що різні значення трактуються як рівні, іноді проти намірів програміста, можуть призвести до проблем із безпекою. Наприклад, результат порівняння `'0e1234' == '0'` є істинним, оскільки рядки, які піддаються аналізу, як числа перетворюються на числа; у цьому випадку перше порівняне значення трактується як наукове позначення, яке має значення (0×10^{1234}) , яке дорівнює нулю. Такі помилки призвели до вразливості аутентифікації у форумі простих машин Туро3 та phpBB при порівнянні хешів паролів MD5. Рекомендованим способом є використання `hash_equals()` (для безпеки часу атаки), `strcmp` або оператора ідентифікації (`===`), оскільки `'0e1234' === '0'` призводить до помилки. [9]

В аналізі 2013 року на понад 170 000 фрагментів веб-сайтів, опублікованому Zone-H, найбільш часто (53%) використовуваною технікою було використання вразливості включення файлів, здебільшого пов'язаного з небезпечним використанням мовних конструкцій PHP, включаючи, вимагати та дозволити `_url_fopen`.

2.4 Опис мови JavaScript

Синтаксис JavaScript схожий мову C та її нащадків. Незважаючи на ім'я, мова програмування JavaScript має дуже мало спільного з Java.

У JavaScript успадкування підтримується прототипами, а не класами. У минулому JavaScript, в основному, використовувалася на стороні клієнта. Реалізації JavaScript або ECMAScript також дозволяють застосувати додатки на сервері, наприклад, за допомогою Node.js та як JScript або JScript .NET в середовищі ASP або ASP.NET в Інтернет-інформаційних службах Microsoft. Іншими прикладами серверного програмування JavaScript є POW та Jaxer, які базуються на Mozilla JavaScript (двигуні SpiderMonkey), V8CGI, який базується на впровадженні JavaScript V8 та інтегрований у HTTP-сервер Apache, та QML - розширена форма JavaScript для малювання та анімації.

Мова також використовується як сценарій для ігор та прикладних програм, оскільки ядро мови містить лише декілька об'єктів (та інтерпретатор, необхідний для виконання сценаріїв) сформульованих у JavaScript, може бути відносно невеликим.

Крім того, JavaScript використовується як мова зв'язку в базі даних MongoDB та в мікроконтролерах. [2]

2.5 Система управління базами даних MySQL

2.5.1 Області застосування та використання

Кращим застосуванням MySQL є зберігання даних для веб-служб. MySQL часто використовується разом із веб-сервером Apache або Nginx та

мовою скриптів PHP. Тоді йдеться про стек LAMP (Linux / Apache / MySQL / PHP) або про стек LEMP ("e" означає [e] nginx). Багато веб-сервісів використовують цю архітектуру та, залежно від їх розміру та потреб, оперують великою кількістю серверів MySQL, за допомогою яких обробляється доступ з мережі. MySQL використовується такими сервісами як Flickr, YouTube, Facebook та Twitter. Окрім того, MySQL використовується в багатьох продуктах як вбудована система баз даних. Зараз MySQL все більше замінюється своїм двійником MariaDB. MariaDB сумісний з MySQL, тому бази даних MySQL можна досить легко імпортувати. Засновник MySQL розробив новий форк після придбання MySQL Oracle та розширив його разом із командою колишніх основних розробників MySQL, тому наразі доступні інші можливості. [16]

2.5.2 Структура системи зберігання

MySQL в основному передбачає сервер MySQL, на якому зберігаються дані, і один або кілька клієнтів MySQL, які надсилають запити на сервер, на які відповідає сервер. Стандартний порт для сервера MySQL - 3306 в протоколі управління передачею (TCP).

Можна створити кілька баз даних в системі управління базами даних, сервером MySQL. В базі даних може бути створено кілька таблиць. MySQL практично створює папку на жорсткому диску для кожної бази даних, в якій зберігаються файли для структури та даних окремих таблиць. Точний формат цих файлів залежить від механізму зберігання даних, який використовується для кожної таблиці.

Кожна таблиця може бути різного типу. Тип таблиці визначає, який механізм зберігання даних (підсистема зберігання даних) використовується для запитів до таблиці. Кожна таблиця може містити стовпці, в яких можуть зберігатися дані певного типу (наприклад, цілі числа або рядки). Максимальний розмір таблиць, як правило, обмежений лише операційною системою. [6]

2.5.3 Процес обробки запиту

Клієнт може надсилати запити до бази даних на сервер MySQL. Вони відповідають за обробку кожного запиту максимально ефективно. По-перше, кеш запитів запитується, а якщо результату не існує, запит аналізується, оптимізується та остаточно виконується; потім результат повертається.

Кеш запитів

Для підвищення продуктивності MySQL може зберігати результати запитів у кеші. Якщо пізніше запит буде надісланий ідентичному запиту на сервер без того, як дані в базі даних тим часом змінилися, на нього буде відповідати кеш. Тоді до самої бази даних не потрібно звертатися.

Розбір

Якщо потрібно виконати запит, спочатку перевіряється, чи правильний його синтаксис. Для цього він розбивається на окремі його компоненти. У той же час збирається основна інформація про запит, наприклад тип запиту (SELECT, INSERT, SET або GRANT), відповідні таблиці та вміст пункту WHERE. В кінці цього кроку MySQL знає дерево розбору, яке потім можна оптимізувати. [14]

Оптимізація

Якщо запит синтаксично дійсний, він буде оптимізований далі. Оптимізатор шукає найефективніший спосіб обробки запиту. Основна спроба - зберегти кількість записів даних для читання якомога менше. Це досягається, наприклад, якщо записи потрібно читати з декількох таблиць за допомогою розумної послідовності запитів (JOIN) таблиць; дані, які не потрібні, повністю видаляються з приєднанням. Оптимізатор, серед іншого, враховує, чи є сенс використовувати індекс (і якщо так, то який), щоб знайти потрібні записи даних, чи краще було б замість цього виконувати сканування таблиці, тобто пройти через всю таблицю.

Наявні альтернативи оцінюються оптимізатором. Оцінюється скільки часу знадобиться для обробки кожного можливого варіанта. Тоді фактично виконується альтернатива, яка має найменші витрати.

Роботу оптимізатора можна прослідкувати за допомогою команди MySQL EXPLAIN. Ця команда (за якою слідує конкретний запит) дає вказівці оптимізатору повернути свій план запитів. Як вихід, користувач отримує інформацію про кожен крок, який буде вжитий оптимізатором для відповіді на запит. Таким чином, за допомогою деяких базових знань, ви можете визначити, в якій мірі запит можна оптимізувати, щоб він міг швидше виконуватись. [16]

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Архітектура web-додатку

Даний сайт складається з декількох сторінок, які взаємодіють між собою.

В першу чергу, при відкритті головної сторінки відображається інформація про призначення цього сайту.

Меню web-застосунку складається із трьох пунктів: головної сторінки, каталогу товарів та контактної інформації.

У верхній частині web-додатку знаходиться панель для авторизації та посилання на сторінку реєстрації.

В нижній частині бачимо інформацію про розробника сайту та посилання на сторінку адміністрування.

При переході до каталогу товарів відображається повідомлення, що користувач повинен авторизуватися або зареєструватися. Зареєстрований відвідувач має можливість авторизуватись та переглянути каталог товарів. При відкритті сторінки реєстрації відображається форма, в якій потрібно заповнити обов'язкові поля: логін, пароль, ім'я та адресу електронної пошти. Якщо всі дані коректні то, після натиснення кнопки «Зареєструватися» в верхній частині сайту відображається привітання.

Після входу користувача, в лівій частині вікна сторінки «Каталог» знаходиться панель, де можна обрати категорію і, в залежності від вибору, з'являється список товарів, які належать до неї.

Також є можливість здійснити пошук необхідного запису за назвою або описом. За бажанням, пошук здійснюється за ціною, шляхом введення її діапазону.

При переході на сторінку «Контакти» авторизований користувач бачить контактні телефони та адресу електронної пошти для зв'язку з адміністратором сайту.

Якщо перейти до модуля адміністрування спочатку буде повідомлення про те, що потрібно здійснити вхід з правами адміністратора. Після введення

правильного логіна та пароля відображається повідомлення про успішну авторизацію та відкривається адміністративний модуль. При виборі категорії відкривається таблиця з повною інформацією про всі записи, які в ній знаходяться. Можна додати новий товар, змінити уже існуючий чи видалити будь-який запис.

3.2 База даних та її адміністрування

Створена база даних «diplom», до якої додано чотири таблиці.

Структура бази даних представлена на рис. 3.1 зображено зв'язок між таблицями «category» (в якій представлені категорії) та «product» (з каталогом товарів) і рис. 3.2. з таблицями «user» та «admin» з даними про користувачів.

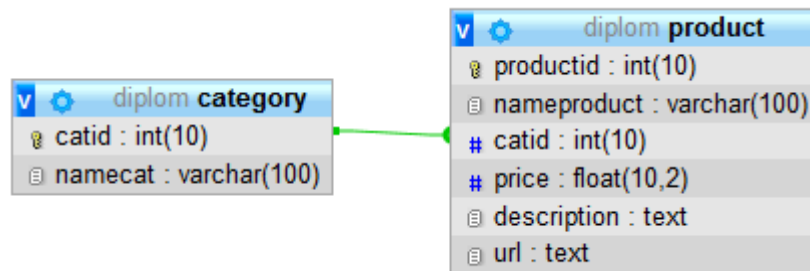


Рисунок 3.1 – Таблиці «category» й «product»

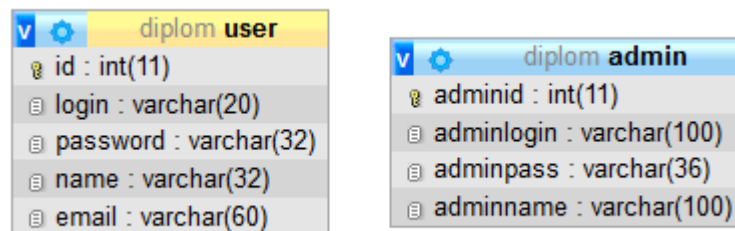


Рисунок 3.2 – Таблиці «user» й «admin»

Структура таблиць:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	catid	int(10)			Нет	Нет		AUTO_INCREMENT
2	namecat	varchar(100)	utf8_general_ci		Нет	Нет		

Рисунок 3.3 – Структура таблиці «category»

Програмний код створення:

```
create table category
(
catid int(10) unsigned not null auto_increment primary key,
catname char(100) not null
);
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	productid	int(10)			Нет	Нет		AUTO_INCREMENT	
2	nameproduct	varchar(100)	utf8_general_ci		Да	NULL			
3	catid	int(10)			Нет	0			
4	price	float(10,2)			Нет	0.00			
5	description	text	utf8_general_ci		Да	NULL			
6	url	text	utf8_general_ci		Да	NULL			

Рисунок 3.4 – Структура таблиці «product»

Програмний код створення:

```
CREATE TABLE `product` (
`productid` int(10) unsigned NOT NULL auto_increment,
`nameproduct` varchar(100) default NULL,
`catid` int(10) unsigned NOT NULL default '0',
`price` float(10,2) NOT NULL default '0.00',
`description` text, PRIMARY KEY (`productid`),
KEY `product_category` (`catid`) ) ;
```

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	Нет		AUTO_INCREMENT
login	varchar(20)	utf8_general_ci		Нет	Нет		
password	varchar(32)	utf8_general_ci		Нет	Нет		
name	varchar(32)	utf8_general_ci		Нет	Нет		
email	varchar(60)	utf8_general_ci		Нет	Нет		

Рисунок 3.5 – Структура таблиці «user»

Програмний код створення:

```
create table user
(id int(11) unsigned not null auto_increment primary key,
login varchar(20) not null,
password varchar(32) not null,
name varchar(32) not null,
email varchar(60) not null);
```


adminid	int(11)			Нет	Нет		
adminlogin	varchar(100)	utf8_general_ci		Нет	Нет		
adminpass	varchar(36)	utf8_general_ci		Нет	Нет		
adminname	varchar(100)	utf8_general_ci		Нет	Нет		

Рисунок 3.6 – Структура таблиці «admin»

Програмний код створення:

```
create table admin
(adminid int(11) unsigned not null primary key,
adminlogin varchar(100) not null,
adminpass varchar(36) not null,
adminname varchar(100) not null);
```

Для редагування та додавання інформації використовується декілька сценаріїв.

Модуль адміністрування взаємодіє з базою даних та створений для додавання, видалення або зміни записів, які знаходяться в таблиці «product».

Зовнішній вигляд модуля адміністрування зображено на рис. 3.7.

Після вибору категорії відображається повна інформація із таблиці «product» (інформація про товари, включаючи: номер запису, назву, номер категорії, ціну, детальний опис та посилання на фото).

Справа від таблиці знаходиться об'єкт типу select де можна здійснити вибір запису за номером та редагувати його.

Під таблицею є кнопка для видалення обраного запису із таблиці.

На цій сторінці знаходиться ще дві форми: перша призначена для додавання запису; друга відповідає за редагування обраного товару.

Обов'язковим полем для заповнення є тільки назва, інші поля можна не заповнювати.

Програмний код розбитий на функції для швидкого оновлення даних таблиці після редагування.

Адміністрування

Оберіть категорію

Ключ	Назва	Категорія	Ціна	Детальна інформація	Посилання на фото
4	Моноблок Vinga AIO BUSINESS B400WP (A22N4200.464WP)	2	9500.00	Комп'ютер • Моноблок • Intel Pentium N4200 • 1,1 ГГц • ОЗУ: 4 ГБ • DDR3 • SSD: 64 (eMMC) ГБ • графічний адаптер: інтегрований Intel HD Graphics 505 • Wi-Fi 802.11 b / g / n / ac, Bluetooth 4.0 • екран: 21,5, IPS, 1920x1080	URL
5	Неттоп Apple Mac mini Late 2018 (MRTT2)	2	36000.00	В середині неттопа Apple Mac mini Late 2018 (MRTT2) 6-ядерний процесор Intel Core i5 з частотою 3,0 ГГц в базовому режимі і 4,1 ГГц в Turbo Boost і інтегрованим графічним чіпом Intel UHD Graphics 630. З пам'яті в пристрої є 256 ГБ основний і 8 Гб оперативної DDR4 SO-DIMM 2666 МГц, а панель інтерфейсів хизується чотирма портами Thunderbolt 3 (USB-C), двома USB 3.0 портами (до 5 Гбіт / с), портом HDMI 2.0 з підтримкою багатоканального аудіовиходу, портом Gigabit Ethernet (налаштовується для 10Gb Ethernet) і роз'ємом для навушників 3,5 мм. Але, крім цього неттоп Apple Mac mini Late 2018 (MRTT2) дозволяє використовувати Bluetooth 5.0 і Wi-Fi 802.11ac сумісний зі стандартами IEEE 802.11a / b / g / n, а сама «коробочка» займає простір 197 x 197 x 36 мм.	URL
6	Десктоп Lenovo Legion C530 (90L20039UL)	2	23300.00	Комп'ютер • Десктоп • Intel Core i5-9400 • 2,9 ГГц • ОЗП: 8 ГБ • DDR4-2666 • HDD: 1000 ГБ • SSD: 128 ГБ • графічний адаптер: NVIDIA GeForce GTX 1650 • 4 ГБ • 802.11 a / b / g / n / ac, Bluetooth 4.1	URL
7	Неттоп HP 260 G3 DM (5JP23ES)	2	6000.00	Комп'ютер • Неттоп • Intel Celeron 3865U • 1,8 ГГц • ОЗП: 4 ГБ • DDR4-2133 • SSD: 128 ГБ • графічний адаптер: інтегрований Intel HD Graphics 610	URL
8	Barebone-неттоп Intel Nuc (BOXNUC7I5BNH)	2	11600.00	Із встановлених компонентів barebone-неттопа Intel Nuc (BOXNUC7I5BNH) є тільки материнська плата з двоядерним процесором Core i5-7260U 2.2 - 3.4 ГГц, графічним адаптером Intel Iris Plus 640, модулями Wi-Fi 802.11ac і Bluetooth 4.2, а також 65 Вт блоком живлення. Окремо встановлювати користувачеві доведеться оперативну пам'ять, якій, в два DDR4 SO-DIMM слота, влізе до 32 ГБ з частотою до 2133 МГц, а також накопичувачі інформації. Для них передбачена підтримка твердотільних накопичувачів з інтерфейсом M.2 тип M (22 x 42 або 22 x 80) і 2.5 "SATA III HDD або SSD (товщиною до 9,5 мм). У свою чергу зовні нічого додавати не потрібно - виробник подбав про наявність на задній панелі 2 x USB 3.0 портів, 1 x Thunderbolt / USB Type C, 1 x RJ-45 LAN, 1 x HDMI і роз'єму живлення, на передній 2 x USB 3.0, один з яких для призначений для зарядки і роз'єму для підключення навушників / мікрофона, а на бічній панелі роз'єм для замка Kensington і кардрідер для microSD карт пам'яті. Також, продумуючи комплектацію, розробники barebone-неттопа Intel Nuc (BOXNUC7I5BNH) поклали в коробку поставки кронштейн VESA з кріпильними гвинтами, блок живлення та документацію.	URL

-
4
5
6
7
8

При видаленні записів будьте обережні. Відновлення неможливе!!! [Видалити](#)

Додати товар

Категорія:

Назва:

Ціна:

Опис:

Посилання на фото:

[Додати](#) [Очистити](#)

Змінити запис

Категорія:

Номер:

Назва:

Ціна:

Опис:

Посилання на фото:

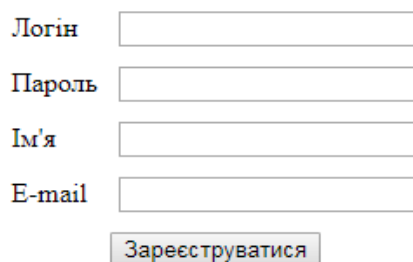
[Зберегти](#) [Очистити](#)

Рисунок 3.7 – Зовнішній вигляд адміністративного модуля

На рис. 3.8 зображена форма, яка призначена для додавання записів в таблицю «user» (реєстрації користувача для подальшого входу на сайт).

Всі поля цієї форми є обов'язковими для заповнення на полях «логін» та «пароль» створено обмеження для довжини значень, які вводяться в поля (не менше 6 символів), також в будь-яке поле неможливо ввести пробіл.

Реєстрація



Логін

Пароль

Ім'я

E-mail

Рисунок 3.8 – Форма реєстрації

Після реєстрації також автоматично проходить авторизація.

Відбувається перехід на сторінку привітання.

3.3 Сценарії та основні модулі інформаційної системи

Кожна сторінка включає в себе підключення додатків: header.php, footer.html. Також одну із сторінок:

- authorization.php - якщо користувач не авторизований з формою авторизації та посиланням на сторінку реєстрації (рис. 3.11);
- auth.php – якщо користувач авторизований з привітанням та посиланням на сторінку виходу (exit.php) (рис. 3.12);
- adminauth.php – якщо користувач авторизований як адміністратор з привітанням та посиланням на сторінку виходу (exit.php) (рис. 3.13);

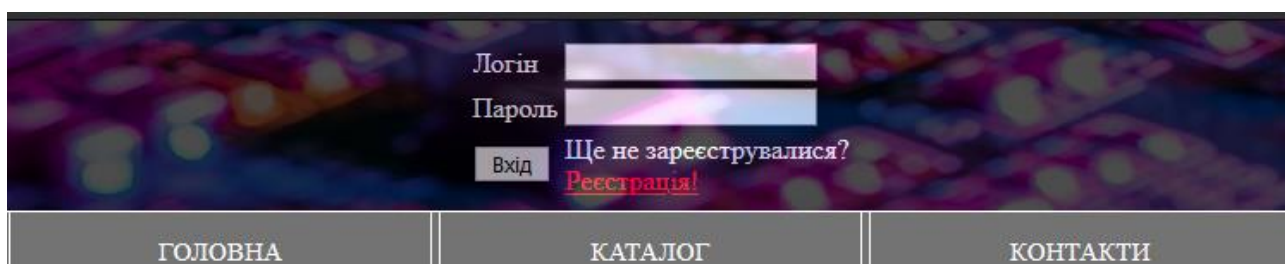


Рисунок 3.9 – Додаток header.php

На цьому додатку знаходиться меню з посиланнями на такі сторінки:

- index.php (головна сторінка);
- catalog.php (каталог товарів);
- contacts.php (контактна інформація).



Рисунок 3.10 – Додаток footer.html

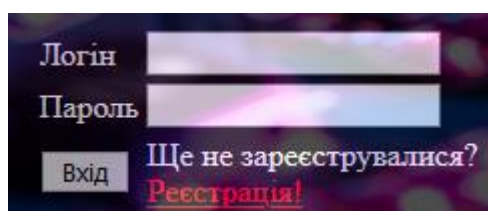


Рисунок 3.11 – Форма authorization.php

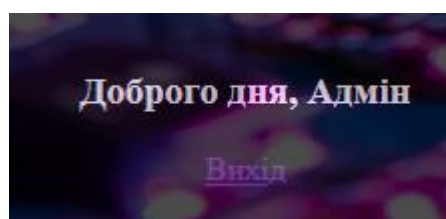


Рисунок 3.12 – Форма auth.php

Вигляд форми для авторизації від імені адміністратора виглядає так само як і форми авторизації звичайного користувача.

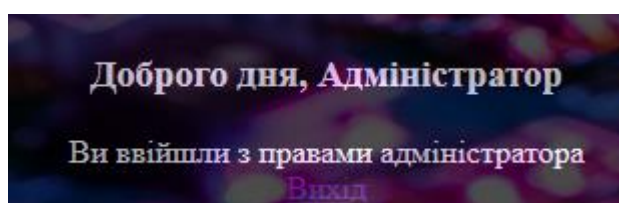
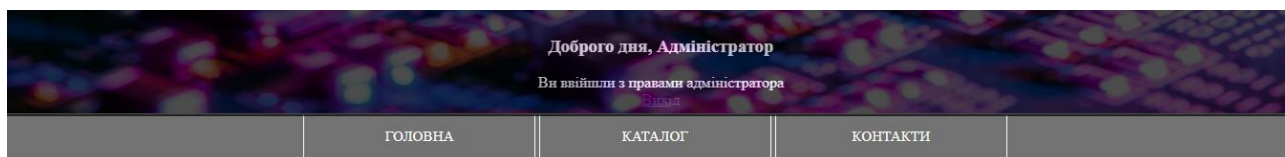


Рисунок 3.13 – Форма adminauth.php

Сторінка exit.php видаляє всі змінні сесії та здійснює переадресацію на головну сторінку.

Головна сторінка відображається відразу після відкриття сайту та містить в собі інформацію про діяльність сайту.



Вітаю на моєму сайті

Рисунок 3.14 – Сторінка index.php

Інформація на сторінці contacts.php доступна тільки для зареєстрованих користувачів та включає в себе контактну інформацію.

Контактна інформація

Контактні телефони:

+380991621651

+380633522183

Електронна адреса:

alexeydanger17@gmail.com

Рисунок 3.15 – Сторінка contacts.php

3.4 Відображення результатів роботи Web-застосунку

Сторінка де відображаються товари (catalog.php) доступна тільки для зареєстрованих користувачів. Після авторизації каталог можна переглянути.

В лівій частині знаходиться меню для вибору категорії, який представляє із себе меню типу select інформація на сторінці відображається відразу після вибору пункту.

Нижче знаходиться форма для пошуку товарів за назвою або описом. Є можливість обрати пошук всіх слів або шукати кожне слово.

Також можна знайти товар заповнивши поля пошуку діапазоном ціни.

Кожен товар показано в окремій комірці з відображенням назви, ціни, опису та фото, яке можна переглянути натиснувши на його зменшену копію.

<p>Всі категорії</p> <p>Ноутбуки</p> <p>Комп'ютери</p> <p>Мобільні телефони</p> <p>Смартфони</p> <p>Планшети</p>	<h2 style="text-align: center;">Асортимент товарів</h2> <div data-bbox="544 383 804 680"></div> <div data-bbox="820 412 1497 651"><p style="text-align: center;">Моноблок Vinga AIO BUSINESS B400WP (A22N4200.464WP)</p><p style="text-align: center;">9500.00 грн.</p><p>Комп'ютер • Моноблок • Intel Pentium N4200 • 1,1 ГГц • ОЗУ: 4 ГБ • DDR3 • SSD: 64 (eMMC) ГБ • графічний адаптер: інтегрований Intel HD Graphics 505 • Wi-Fi 802.11 b / g / n / ac, Bluetooth 4.0 • екран: 21,5, IPS, 1920x1080</p></div>
--	---

ВИСНОВКИ

В роботі було проаналізовано методи методологією розроблення інформаційного та програмного забезпечення інформаційної системи на основі використання баз даних та web-технологій, також ознайомився з сучасними технологіями передачі та обробки інформаційних потоків через Інтернет, архітектурою web-орієнтованих інформаційних систем, новітніми методами розроблення динамічних web-сайтів, які включають в себе такі програмні засоби:

- мова створення веб-додатків HTML;
- мова програмування серверних веб-додатків;
- скриптова мова JavaScript (переважно використовувалася для створення обмежень по вводу значень в поля);
- таблиця стилів CSS;
- засоби управління базами даних та мова програмування для роботи з базами даних MySQL.

В результаті отримано програмне рішення інтернет-магазину, яке може бути впроваджене на підприємстві, якому необхідний каталог інтернет-магазину.

СПИСОК ЛІТЕРАТУРИ

1. Аргерих Л. Профессиональное PHP программирование. - СПб: Символ-Плюс, 2014. - 1048 с.
2. Батурін В. М. Особливості навчання веб-програмування мовою JavaScript студентів-математиків / В. М. Батурін // Вісник Житомирського державного університету. – 2016. - №1(73). - С. 79-83.
3. Веб-дизайн : підруч. для студ. вищ. навч. закл. / О. В. Пасічник, В. В. Пасічник ; за заг. ред. В. В. Пасічника ; М-во освіти і науки України. – Л. : Магнолія 2012, 2015. – 519
4. Гончаров, А.Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник / А.Ю. Гончаров. — КУДИЦ-ПРЕСС, 2015. — 540 с.
5. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - М.: Эксмо, 2015. - 480 с.
6. Дэвис М. Изучаем PHP и MySQL / М. Дэвис, Дж. Филлипс. – М.: Символ-Плюс, 2017. – 442 с.
7. Когтзолл Д. PHP. Полное руководство. — М. : Издательский дом "Вильямс", 2016. - 752 с.
8. Колисниченко Д.М. PHP. – СПб: Наука и техника, 2014. - 567 с.
9. Котеров Д. PHP 5 / Д. Котеров, А. Костарев. – СПб.: БХВ-Петербург, 2017. – 1104 с.
10. Кузнецов М. В., Симдянов И. В., Гольшев С. В. PHP на примерах. — СПб.: БХВ-Петербург, 2015. — 576 с.
11. Кузнецов М. В., Симдянов И. В., Гольшев С. В. PHP. Практика разработки Web-сайтов. — СПб.: БХВ-Петербург, 2015. — 960 с
12. Кузнецов М.В. PHP. Практика создания Web-сайтов / М. В. Кузнецов, И. В. Симдянов. – СПб.: БХВ-Петербург, 2018. – 1244 с.
13. Ленгсторф Дж. PHP и jQuery для профессионалов / Дж. Ленгсторф. – СПб.: Вильямс, 2017. – 334 с.

14. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и javascript / Р. Никсон. – СПб.: Питер, 2011. – 496 с.
15. Пасічник О. В. Веб-дизайн: підручник / О. В. Пасічник, В. В. Пасічник. – Львів: Магнолія 2006. - 2010. – 520 с.
16. Суэринг С. PHP и MySQL. Библия программиста / С. Суэринг, Т. Конверс, Дж. Парк. – СПб.: Диалектика, 2010. – 912 с.
17. Ташков П. А. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX / П. А. Ташков. - СПб. : Питер, 2010. —512 с.
18. Ульман Л. Основы программирования на PHP. - М.: ДМК Пресс, 2001. - 288 с.
19. Веб-сайт: определение и применение [Электронный ресурс]. – Режим доступа: <http://www.webtec.com.ua/articles/index/view/2011-03-31/web-site> –
20. Иван Блинков. Архитектура интерактивных сайтов [Электронный ресурс] /Иван Блинков. – Режим доступа: <http://www.insightit.ru/masshtabiruemost/arkhitektura-interaktivnykh-sajtov/>
21. Язык HTML [Электронный ресурс]. – Режим доступа до ресурсу: <http://net.e-publish.ru/p30aa1.html>.
22. Недостатки PHP. [Электронный ресурс]. – Режим доступа: <http://comphobby.ru/2011/06/07/nedostatkiphp-nedostatki-1-2/>

ДОДАТОК А

<!--admin.php-->

```
<?session_start();
require 'header.php';
if($_SESSION['id']==2){
require 'adminauth.php';
function show(){
error_reporting(0);?>
<body>
<script>$(".mydiv").load(location.href + " .mydiv");</script>
<div class="text">
<h1>Адміністрування</h1>
<h3>Оберіть категорію</h3>
<form name="search" method="post">
<select name="adminsel" id="adminsel"
onchange="this.form.submit()" method="post" size="1">
<option value='0'<?php if($_POST['adminsel'] == '0') echo
"selected"; ?>> - </option>
<option value='1'<?php if($_POST['adminsel'] == '1') echo
"selected"; ?>>Ноутбуки</option>
<option value='2'<?php if($_POST['adminsel'] == '2') echo
"selected"; ?>>Комп'ютери</option>
<option value='3'<?php if($_POST['adminsel'] == '3') echo
"selected"; ?>>Мобільні телефони</option>
<option value='4'<?php if($_POST['adminsel'] == '4') echo
"selected"; ?>>Смартфони</option>
<option value='5'<?php if($_POST['adminsel'] == '5') echo
"selected"; ?>>Планшети</option>
</select>
<?php
$db = mysql_connect("localhost","root","");
$a=mysql_select_db("diplom",$db);
for($i=1;$i<=5;$i++){
if($_POST['adminsel']==$i){
if ($a) {echo "";} else {echo "error!";}
$ggg= "SELECT * From product WHERE catid='$i'";
$z=$i;
$del=array();
$result = mysql_query ($ggg,$db);
if ($result) {echo "";} else {echo "er-ror!";} echo "<table
border=1 width=95% cellpadding=5 cellspacing=0>";
$p=1;
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {
if ($p == 1) {
echo"<tr bgcolor='#aaa' align='center'>";
echo "<td>Ключ</td>";
echo "<td>Назва</td>";
echo "<td>Категорія</td>";
echo "<td>Ціна</td>";
echo "<td>Детальна інформація</td>";
```

```

echo "<td>Посилання на фото</td>";
echo"</tr>";
$p=2;}
$x=1;
echo"<tr>";
foreach ($myrow as $key => $buf) {
$key=$x;
$x++;
if($key==1) {
$del[]=$buf;
echo"<td align='center'>$buf</td>";}
if($key==2) {
echo"<td>$buf</td>";}
if($key==3) {
echo"<td align='center'>$buf</td>";}
if($key==4) {
echo"<td align='center'>$buf</td>";}
if($key==5) {
echo"<td>$buf</td>";}
if($key==6) {
if($buf==NULL) {echo"<td align='center'></td>";}
else {echo"<td align='center'><a href='$buf'
target='_blank'>URL</a></td>";}}}
echo"</tr>";}
echo "</table>";
$N=count($del)+1;
echo"<div id='seldel'><select name='admindel' method='post'
onchange='this.form.submit()' size='$N' multiple>";
?><option value='999999999'<? if(isset($_POST['delete'])) echo
"selected";?>> - </option><?
foreach($del as $key => $value) {
?><option value='<?echo $key;?>'<? if($_POST['admindel'] == $key)
echo "selected"; ?>> <?echo $value;?> </option><?>
echo'</div></select>';
for($k=0;$k<=$N;$k++){
if($_POST['admindel']==$k) {
$ch=$del[$k];}
if($_POST['admindel']==999999999) {
$ch=NULL;}}
echo'<script type="text/javascript">this.form.submit();</script>'
?><script type="text/javascript">
function proverka(input) {
input.value = input.value.replace(/[\^\d.]/g, '');};
</script>
</div>
<p align="right">При видаленні записів будьте обережні.
Відновлення неможливе!!!
<input name="delete" id="delete" type="submit"
value="Видалити"/></p>
<h3>Додати товар</h3>
<table><tr><td>Категорія:</td>

```

```

<td colspan="2"><input type="text" readonly="true" name="catid"
value="<?echo $z;?>"></td></tr>
<tr><td>Назва:</td>
<td colspan="2"><input maxlength="100" name="nameproduct"
id="nameproduct" size="20" type="text" /></td></tr>
<tr><td>Ціна:</td>
<td colspan="2"><input maxlength="9" onkeyup="return
proverka(this);" name="price" min="1" max="999999999" size="20"
type="text" /></td></tr>
<tr><td>Опис:</td>
<td colspan="2"><textarea cols="39" maxlength="10000"
name="description" rows="5"></textarea></td></tr>
<tr><td>Посилання на фото:</td>
<td colspan="2"><input name="url" size="37" type="text"
/></td></tr>
<tr><td></td>
<td width="95px"><input name="add" type="submit" value="Додати"
/></td>
<td><input name="reset" type="reset" value="Очистити"
/></td></tr></table>
<? $db = mysql_connect("localhost","root","");
$users=mysql_select_db("diplom", $db);
$nh= "SELECT nameproduct From product WHERE productid='$sch'";
$nh1 = mysql_query ($nh,$db);
$nameproductc = mysql_fetch_array($nh1,MYSQL_ASSOC);
if($nameproductc!=NULL) {
foreach($nameproductc as $nameproductch){}}
$pr= "SELECT price From product WHERE productid='$sch'";
$pr1 = mysql_query ($pr,$db);
$pricec = mysql_fetch_array($pr1,MYSQL_ASSOC);
if($pricec!=NULL) {
foreach($pricec as $pricech){}}
$ds= "SELECT description From product WHERE productid='$sch'";
$ds1 = mysql_query ($ds,$db);
$descriptionc = mysql_fetch_array($ds1,MYSQL_ASSOC);
if($descriptionc!=NULL) {
foreach($descriptionc as $descriptionch){}}
$u= "SELECT url From product WHERE productid='$sch'";
$ur = mysql_query ($u,$db);
$urlc = mysql_fetch_array($ur,MYSQL_ASSOC);
if($urlc!=NULL) {
foreach($urlc as $urlch){}}?>
<br><h3>Змінити запис</h3>
<table><tr><td>Категорія:</td>
<td colspan="2"><input type="text" readonly="true" name="catidch"
value="<?echo $z;?>"></td></tr>
<tr><td>Homep:</td>
<td colspan="2"><input type="text" readonly="true"
name="productidch" value="<?echo $sch;?>"></td></tr>
<tr><td>Назва:</td>

```

```

<td colspan="2"><input maxlength="100" name="nameproductch"
id="nameproductch" size="20" type="text" value="<?echo
$nameproductch;?>" /></td></tr>
<tr><td>Ціна:</td>
<td colspan="2"><input maxlength="9" onkeyup="return
proverka(this);" name="pricech" min="1" max="999999999" size="20"
type="text" value="<?echo $pricech;?>" /></td></tr>
<tr><td>Опис:</td>
<td colspan="2"><textarea cols="39" maxlength="10000"
name="descriptionch" rows="5"><?echo
$descriptionch;?></textarea></td></tr>
<tr><td>Посилання на фото:</td>
<td colspan="2"><input name="urlch" size="37" type="text"
value="<?echo $urlch;?>" /></td></tr>
<tr><td></td><td width="95px"><input name="changech" type="submit"
value="Зберегти" /></td>
<td><input name="reset" type="reset" value="Очистити"
/></td></tr></table></form>
<?php
}}
function delett(){
if (!empty($_POST)) {
$productiddel = $_POST['productidch']; }
$host = 'localhost';
$user = 'root';
$pass = '';
$db_name = 'diplom';
$link = mysqli_connect($host, $user, $pass, $db_name);
if(isset ($_POST['delete'])){
$query = "DELETE FROM product WHERE productid = '$productiddel'";
$result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
mysqli_close($link);}}
function change(){
if (!empty($_POST)) {
$productidch1 = $_POST['productidch']; }
if (!empty($_POST)) {
$nameproductch1 = $_POST['nameproductch']; }
if (!empty($_POST)) {
$catidch1 = $_POST['catidch']; }
if (!empty($_POST)) {
$pricech1 = $_POST['pricech']; }
if (!empty($_POST)) {
$descriptionch1 = $_POST['descriptionch']; }
if (!empty($_POST)) {
$urlch1 = $_POST['urlch']; }
$host = 'localhost';
$user = 'root';
$pass = '';
$db_name = 'diplom';
$link = mysqli_connect($host, $user, $pass, $db_name);
if(isset ($_POST['changech'])){

```

```

if($nameproductch1!=''){
if($productidch1!=''){
if (!$link) {
echo "З'єднання неможливе" . mysqli_connect_errno() . ", помилка:
" . mysqli_connect_error();
exit;}
$sql = mysqli_query($link, "UPDATE `product` SET `nameproduct` =
'$nameproductch1', `catid` = '$catidch1', `price` = '$pricech1',
`description` = '$descriptionch1' , `url` = '$urlch1' WHERE
`product`.`productid` = $productidch1;");
if ($sql) {}
else {echo "<p>Виникла помилка " . mysqli_error($link) . "</p>";}}
else{echo"<p><font color=red>Спочатку оберїть
запис!</font></p>";}}
else{echo"<p><font color=red>Потрїбно ввести
назву!</font></p>";}}
function add(){
if (!empty($_POST)) {
$nameproduct = $_POST['nameproduct']; }
if (!empty($_POST)) {
$catid = $_POST['catid']; }
if (!empty($_POST)) {
$price = $_POST['price']; }
if (!empty($_POST)) {
$description = $_POST['description']; }
if (!empty($_POST)) {
$url = $_POST['url']; }
$host = 'localhost';
$user = 'root';
$pass = '';
$db_name = 'diplom';
$link = mysqli_connect($host, $user, $pass, $db_name);
if(isset ($_POST['add'])) {
if($nameproduct!=''){
if (!$link) {
echo "З'єднання неможливе" . mysqli_connect_errno() . ", помилка:
" . mysqli_connect_error();
exit;}
$sql = mysqli_query($link, "INSERT INTO `product` (`productid`,
`nameproduct`, `catid`, `price`, `description`, `url`) VALUES
(NULL, '$nameproduct', '$catid', '$price', '$description',
'$url');");
if ($sql) {}
else {echo "<p>Виникла помилка " . mysqli_error($link) . "</p>";}}
else{echo"<p><font color=red>Потрїбно ввести
назву!</font></p>";}}
change(); delect(); add(); show();}
else{require 'adminauthorization.php';
echo'<h1>Для перегляду цієї сторїнки необхідно </h1><h1>здїйснити
вхїд з правами адмінїстратора</h1>';?>
</div></body></html>
<?require 'footer.html';?>

```

ДОДАТОК Б

<!--adminauth.php-->

```
<?session_start();?>
<div id="logina"><meta charset="\utf-8\" /><h3><p><center><font
color="white">Доброго дня,
<?error_reporting(0);
echo $_SESSION['adminlogin'];?></p>
</h3><center>Ви ввійшли з правами адміністратора</font>
<center><a href="exit.php">Вихід</a></center></div>
```

ДОДАТОК В

<!--adminauthorization.php-->

```
<?session_start();
error_reporting(0);
if (!empty($_POST)) {
$adminlogin = $_POST['adminlogin']; }
if (!empty($_POST)) {
$password = $_POST['adminpass'];
$adminpass = md5($password);}
$db = mysql_connect("localhost","root","");
$user=mysql_select_db("diplom", $db);
$l= "SELECT adminid From admin WHERE adminlogin='$adminlogin' AND
adminpass='$adminpass'";
$r = mysql_query ($l,$db);
$r1 = mysql_fetch_array($r,MYSQL_ASSOC);
if($r1!=NULL){foreach($r1 as $r2){}}
if (!isset($adminlogin)&&!isset($adminpass)){?>
<meta charset="utf-8">
<link href="stylez.css" rel="stylesheet"><center>
<div class="containerlogin">
<div id="login"><form method=post action="admin.php">
<table><tr><td><label for="user_adminlogin"><font
color="white">Логін</font></td>
<td><input type=text size=20 name=adminlogin></label></td></tr>
<tr><td><label for="user_pass"><font
color="white">Пароль</font></td>
<td><input type=password size=20 name=adminpass></label></td></tr>
<tr><td><input name="login" type="submit" value="Вхід" /></td>
<td><font color="white">Ще не зареєструвалися?</font><br><a href=
"register.php"><font
color="red">Реєстрація!</font></a></td></tr></table></form></div><
/div></center>
<?>else if(isset($adminlogin)&&isset($adminpass)){
if($r2==NULL){
echo "<div id=\"login\"><meta charset=\"utf-8\"
/><center>Неправильний логін чи пароль. <br>Спробуйте ще
раз<br><table cellpadding='10px'><tr><td><a
href=\"admin.php\"><<Повернутися</a></td></tr></table><center></di
v>";}
if($r2!=NULL){
$e= "SELECT adminname From admin WHERE adminlogin='$adminlogin'";
$er = mysql_query ($e,$db);
$er1 = mysql_fetch_array($er,MYSQL_ASSOC);
if($er1!=NULL){foreach($er1 as $er2){}}
$_SESSION['adminlogin']=$er2;
$_SESSION['id']=2;
require 'adminauth.php';
echo "<script language='JavaScript'
type='text/javascript'>window.location.reload()</script>";}}?>
```


ДОДАТОК Г

<!--auth.php-->

```
<?session_start();  
error_reporting(0);  
echo "<div id=\"login\"><meta charset=\"utf-8\"  
></div><h3><center><font color=\"white\">Доброго дня, ";  
echo $_SESSION['login'];  
echo '</font></center></h3><center><a  
href="exit.php">Вихід</a></center></div>';?>
```

ДОДАТОК Г

<!--authorization.php-->

```
<?session_start();
error_reporting(0);
if (!empty($_POST)) {
$login = $_POST['login']; }
if (!empty($_POST)) {
$password = $_POST['password'];
$password = md5($password);}
$db = mysql_connect("localhost","root","");
$user=mysql_select_db("diplom", $db);
$l= "SELECT id From user WHERE login='$login' AND
password='$password'";
$r = mysql_query ($l,$db);
$r1 = mysql_fetch_array($r,MYSQL_ASSOC);
if($r1!=NULL){
foreach($r1 as $r2){}
if (!isset($login)&&!isset($password)) {?>
<html>
<meta charset="utf-8">
<link href="stylez.css" rel="stylesheet">
<center><div class="containermlogin"><div id="login">
<form method=post action="index.php">
<table><tr><td><label for="user_login"><font
color="white">Логін</font></td>
<td><input type=text size=20 name=login></label></td></tr>
<tr><td><label for="user_pass"><font
color="white">Пароль</font></td>
<td><input type=password size=20 name=password></label></td></tr>
<tr><td><input name="logg" type="submit" value="Вхід" /></td>
<td><font color="white">Ще не зареєструвалися?<br></font><a href=
"register.php"><font
color="red">Реєстрація!</font></a></td></tr></table></form></div><
/div></center>
<?>else if(isset($login)&&isset($password)) {
if($r2==NULL){
echo "<div id=\"login\"><meta charset=\"utf-8\"
/><center>Неправильний логін чи пароль. <br>Спробуйте ще раз або
зареєструйтеся<br><table cellpadding='10px'><tr><td><a
href=\"index.php\"><<Повернутися</a></td><td><a href=
\"register.php\">Реєстрація!</a></td></tr></table><center></div>";
} if($r2!=NULL){
$e= "SELECT name From user WHERE login='$login'";
$r = mysql_query ($e,$db);
$r1 = mysql_fetch_array($r,MYSQL_ASSOC);
if($r1!=NULL){
foreach($r1 as $r2){}
$_SESSION['login']=$r2;
$_SESSION['id']=1;
require 'auth.php';}}?></html>
```

ДОДАТОК Д

<!--catalog.php-->

```
<?session_start();?><html>
<?function catalog(){
error_reporting(0);?>
<div class="text">
<link href="stylez.css" rel="stylesheet">
<form name="search" method="post">
<select id="selcat" name="sel" onchange="this.form.submit()"
method="post" size="7">
<option value='7'<?php if($_POST['sel'] == '7') echo "selected";
?>>Всі категорії</option>
<option value='1'<?php if($_POST['sel'] == '1') echo "selected";
?>>Ноутбуки</option>
<option value='2'<?php if($_POST['sel'] == '2') echo "selected";
?>>Комп'ютери</option>
<option value='3'<?php if($_POST['sel'] == '3') echo "selected";
?>>Мобільні телефони</option>
<option value='4'<?php if($_POST['sel'] == '4') echo "selected";
?>>Смартфони</option>
<option value='5'<?php if($_POST['sel'] == '5') echo "selected";
?>>Планшети</option>
<option id="selcat0" value='0'<?if(isset($_POST['search'])) echo
"selected";?>></option>
</select></form><div id="txt"><?
if($_POST['sel'] == 0){echo'<h1>Асортимент товарів</h1>';}
$db = mysql_connect("localhost","root","");
$a=mysql_select_db("diplom", $db);
for($i=1;$i<=5;$i++){if($_POST['sel']==$i){
echo'<h1>Асортимент товарів</h1>';
if ($a) {echo "";} else {echo "error!";}
$ggg= "SELECT `url`,`nameproduct`,`price`,`description` FROM
`product` WHERE catid=$i";
$result = mysql_query ($ggg,$db);
if ($result) {echo "";} else {echo "er-ror!";}
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){$key=$z; $z++;
if($key==1){
echo"<table><tr><td rowspan=\"3\" width=\"120px\">";
if($buf!=NULL){
echo "<a href='$buf' target='_blank'><img src='$buf' alt='
class='leftimg'></a>";}
if($buf==NULL){
echo '';}
echo"</td><td>";}
if($key==2){echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
```

```

if($key==4){echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}
echo"</div>";}}}}
if($_POST['sel']==7){
echo'<h1>Асортимент товарів</h1>';
if ($a) {echo "";} else {echo "error!";}
$ggg= "SELECT `url`, `nameproduct`, `price`, `description` FROM
`product`";
$result = mysql_query ($ggg,$db);
if ($result) {echo "";} else {echo "er-ror!";}
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){
$key=$z; $z++;
if($key==1){
echo"<table><tr><td rowspan=\"3\" width=\"120px\">";
if($buf!=NULL){
echo "<a href='$buf' target='_blank'><img src='$buf' alt=''
class='leftimg'></a>";}
if($buf==NULL){
echo '';}
echo"</td><td>";}
if($key==2){
echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){
echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}
echo"</div>";}}}?>
<div id="search">
<form name="search1" method="post"><table><tr><td>
Пошук по назві та опису
<?if (!empty($_POST)) {
$search = $_POST['string']; }?>
</td></tr><tr><td>
<input type=search name="string" <?echo"value='$search'"?>>
</td></tr><tr><td><select name="logic">
<option value='OR' <?php if($_POST['logic'] == 'OR') echo
"selected"; ?>>шукати будь-яке слово
<option value='AND' <?php if($_POST['logic'] == 'AND') echo
"selected"; ?>>шукати всі слова
</select></td></tr><tr><td>
<input name="search" type="submit" value="Пошук" />
</td></tr></table></form></div>
<?if (!empty($_POST)) {
$search = $_POST['string']; }
if(isset ($_POST['search']))){
echo"<h3>Результат пошуку</h3>";
$db = mysql_connect("localhost","root","");

```

```

$a=mysql_select_db("diplom", $db);
$search = substr($search, 0, 64);
$search = preg_replace("/[^\w\x7F-\xFF\s]/", " ", $search);
$good = trim(preg_replace("/\s(\S{1,2})\s/", " ", ereg_replace("
+", " ", $search )));
if($_POST['logic']=='OR'){
if ($a) {echo "";} else {echo "error!";}
$sss= "SELECT `url`, `nameproduct`, `price`, `description` FROM
`product` WHERE nameproduct LIKE '%" . str_replace(" ", "%' OR
nameproduct LIKE '%", $good). "%' OR description LIKE '%" .
str_replace(" ", "%' OR description LIKE '%", $good). "%'";
$result = mysql_query ($sss,$db);
if ($result) {echo "";} else {echo "er-ror!";}
$x=0;
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){$key=$z; $z++;
if($key==1){$x++; echo"<table><tr><td rowspan=\"3\"
width=\"120px\">";
if($buf!=NULL){echo "<a href='$buf' target='_blank'><img
src='$buf' alt='' class='leftimg'></a>";}
if($buf==NULL){echo '';}
echo"</td><td>";}
if($key==2){echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}
echo"</div>";}
echo"Пошук завершено! Кількість знайдених записів: $x";}
if($_POST['logic']=='AND'){
if ($a) {echo "";} else {echo "error!";}
$sss= "SELECT `url`, `nameproduct`, `price`, `description` FROM
`product` WHERE nameproduct LIKE '%" . str_replace(" ", "%' AND
nameproduct LIKE '%", $good). "%' OR description LIKE '%" .
str_replace(" ", "%' AND description LIKE '%", $good). "%'";
$result = mysql_query ($sss,$db);
if ($result) {echo "";} else {echo "er-ror!";}
$x=0;
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {
$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){$key=$z; $z++;
if($key==1){$x++; echo"<table><tr><td rowspan=\"3\"
width=\"120px\">";
if($buf!=NULL){echo "<a href='$buf' target='_blank'><img
src='$buf' alt='' class='leftimg'></a>";}
if($buf==NULL){echo '';}
echo"</td><td>";}

```

```

if($key==2){echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){
echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}
echo"</div>";}
echo"Пошук завершено! Кількість знайдених записів: $x";}}
?><div id="sprice">
<form name="search2" method="post"><table><tr>
<td colspan="2">Пошук за ціною</td>
<?if (!empty($_POST)) {
$pricemin = $_POST['pricemin']; }
if (!empty($_POST)) {$pricemax = $_POST['pricemax']; }?>
</tr><tr><td>Від:</td>
<td><input type=search onkeyup="return proverka(this);"
name="pricemin" <?echo"value=' $pricemin' "?>></td>
</tr><tr><td>До:</td>
<td><input type=search onkeyup="return proverka(this);"
name="pricemax" <?echo"value=' $pricemax' "?>></td>
</tr><tr><td colspan="2"><input name="sprice" type="submit"
value="Пошук" />
</td></tr></table></form></div>
<?if ($pricemin==NULL){$pricemin=0;}
if ($pricemax==NULL){$pricemax=999999999;}
if(isset ($_POST['sprice']))){echo"<h3>Результат пошуку</h3>";
$db = mysql_connect("localhost","root","");
$a=mysql_select_db("diplom", $db);
if ($a) {echo "";} else {echo "error!";}
$ppp= "SELECT `url`, `nameproduct`, `price`, `description`, `url`
FROM `product` WHERE `price` >= $pricemin AND `price` <=
$pricemax";
$result = mysql_query ($ppp,$db);
if ($result) {echo "";} else {echo "er-ror!";}
$x=0;
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){$key=$z; $z++;
if($key==1){$x++;
echo"<table><tr><td rowspan=\"3\" width=\"120px\">";
if($buf!=NULL){echo "<a href=' $buf' target='_blank'><img
src=' $buf' alt='' class='leftimg'></a>";}
if($buf==NULL){echo '';}
echo"</td><td>";}
if($key==2){echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}

```

```
echo"</div>";}
echo"Пошук завершено! Кількість знайдених записів: $x";}}
?><script type="text/javascript">
function proverka(input) {
input.value = input.value.replace(/[^\d]/g, '');
};
</script><?
require 'header.php';
if($_SESSION['id']==1){require 'auth.php';
catalog();}
else if($_SESSION['id']==2){
require 'adminauth.php';
catalog();}
else{require 'authorization.php';
echo'<h1>Ця сторінка доступна тільки для зареєстрованих
користувачів</h1>';}??
</div></div></body></html>
<?require 'footer.html';?>
```

ДОДАТОК Е

<!--contact.php-->

```
<?session_start();
?><html><?
error_reporting(0);
require 'header.php';
function contact(){
echo'<div class="text"><h1>Контактна інформація</h1>
<table><tr>
<td>Контактні телефони:</td></tr><tr>
<td>+380991621651</td></tr>
<tr><td>+380633522183</td></tr><tr><td><br></td></tr><tr>
<td>Електронна адреса:</td></tr><tr>
<td>alexeydanger17@gmail.com</td></tr></table></div>';
}
if($_SESSION['id']==1){
require 'auth.php';
contact();
}
else if($_SESSION['id']==2){
require 'adminauth.php';
contact();
}
else{
require 'authorization.php';
echo'<h1>Ця сторінка доступна тільки для зареєстрованих
користувачів</h1>';
}
?>
<?require 'footer.html';?>
</html>
```


ДОДАТОК Є

<!--exit.php-->

```
<?session_start();
error_reporting(0);
session_unset();
session_destroy();
require 'index.php';?>
```

ДОДАТОК Ж

<!--footer.html-->

```
<html><footer><a href="admin.php">Адміністрування</a>
<p>Виконав студент групи ІНЗ-61с Кременчуков
О.В.</p></footer></html>
```

ДОДАТОК З

<!--header.php-->

```
<link href="stylez.css" rel="stylesheet">
<meta charset="utf-8">
<title>Мій сайт</title>
<div class="head">
</div>
<body><div class="ul"><ul class="menu-main">
<li><a href="index.php">ГОЛОВНА</a></li>
<li><a href="catalog.php">КАТАЛОГ</a></li>
<li><a href="contact.php">КОНТАКТИ</a></li></ul></div>
```

ДОДАТОК И

<!--index.php-->

```
<?session_start();
error_reporting(0);
require 'header.php';
if($_SESSION['id']==1){
require 'auth.php';}
else if($_SESSION['id']==2){
require 'adminauth.php';}
else{require 'authorization.php';}?>
<html><head><div class="text">
<h1>Вітаю на моєму сайті</h1>
</div></body></html>
<?require 'footer.html';?>
```

ДОДАТОК I

<!--reged.php-->

```
<?session_start();  
error_reporting(0);?>  
<html><head><div class="text">  
<h1>Ви успішно зареєстровані!</h1>  
</div></body></html>
```

ДОДАТОК II

<!--register.php-->

```
<?session_start();  
error_reporting(0);  
require 'header.php';  
require 'registration.php';  
if($_SESSION['id']==1){  
require 'auth.php';}  
else{require 'authorization.php';}?>  
<html><head><body></body></html>  
<?require 'footer.html';?>
```

ДОДАТОК Й

<!--registration.php-->

```
<?session_start();
error_reporting(0);
if (!empty($_POST)) {
$login = htmlspecialchars($_POST['login']);}
if (!empty($_POST)) {
$password = htmlspecialchars($_POST['password']);}
$password = md5($password);}
if (!empty($_POST)) {
$name = htmlspecialchars($_POST['name']);}
if (!empty($_POST)) {
$email = htmlspecialchars($_POST['email']);}?>
<script>
function check(input) {
input.value = input.value.replace(/[ ]/g, '');};
</script>
<link href="stylez.css" rel="stylesheet">
<div class="text">
<div class="reg">
<h1>Рееcтpация</h1>
<form method=post>
<table cellpadding="" cellspacing="10">
<tr><td>Лorин</td>
<td><input name="login" id="logginn" size="20" maxlength="20"
onkeyup="return check(this)" type="text" value="<?echo
$login;?>" /></td></tr>
<tr><td>Пapоль</td>
<td><input name="password" size="20" maxlength="15"
onkeyup="return check(this)" type="password" /></td></tr>
<tr><td>Им'я</td>
<td><input name="name" size="20" maxlength="32" onkeyup="return
check(this)" type="text" value="<?echo $name;?>" /></td></tr>
<tr><td>E-mail</td>
<td><input name="email" size="20" maxlength="60" onkeyup="return
check(this)" type="email" value="<?echo $email;?>" /></td></tr>
<tr><td colspan="2" style="text-align: center;">
<input name="regist" type="submit"
value="Зapeecтpувaтися" /></td></tr></table></form></div>
<?$host = 'localhost';
$user = 'root';
$password = '';
$db_name = 'diplom';
$link = mysqli_connect($host, $user, $pass, $db_name);
if(isset ($_POST['regist'])) {
if($login!=""&&strlen($login)>=6) {
if($password!=""&&strlen($password)>=6) {
if($name!="") {
if($email!="") {
$db = mysql_connect("localhost","root","");
```

```

$a=mysql_select_db("diplom", $db);
$ggg= "SELECT * From user WHERE login='$login'";
$result = mysql_query($ggg,$db);
$myrow = mysql_fetch_array($result,MYSQL_ASSOC);
if($myrow==NULL){
if (!$link) {
echo "Невдача: " . mysqli_connect_errno() . ", помилка: " .
mysqli_connect_error();
exit;}
$sql = mysqli_query($link, "INSERT INTO `user` (`id`, `login`,
`password`, `name`, `email`) VALUES (NULL, '$login', '$password',
'$name', '$email');");
if ($sql) {
echo'<style type="text/css">.reg{opacity: 0;}</style>';
require 'reged.php';}
else {
echo "<p>Виникла помилка " . mysqli_error($link) . "</p>";}}
else{echo'<center>Користувач з таким логіном уже
zareestrovaniy!</center>';}}
else{echo"<font color=red>Потрібно ввести електронну
adresu!</font>";}}
else{echo"<font color=red>Потрібно ввести ім'я!</font>";}}
else{echo"<font color=red>Потрібно ввести пароль або він
zakortkiy (ne menше 6 simvoliv)!</font>";}}
else{echo"<font color=red>Потрібно ввести логін або він zakortkiy
(ne menше 6 simvoliv)!</font>";}}?>
</div>

```

ДОДАТОК К

```
<!--search.php-->
<link href="stylez.css" rel="stylesheet">
<meta charset="utf-8">
<div class="text">
<div id="search">
<form name="search1" method="post">
<table><tr><td>
<?if (!empty($_POST)) {
$search = $_POST['string'];
}error_reporting(0);?>
<input type=search name="string" <?echo"value='.$search.'"?>>
</td></tr><tr><td>
<select name="logic">
<option value='OR' <?php if($_POST['logic'] == 'OR') echo
"selected"; ?>>шукати будь-яке слово
<option value='AND' <?php if($_POST['logic'] == 'AND') echo
"selected"; ?>>шукати всі слова
</select>
</td></tr><tr><td>
<input name="search" type="submit" value="Пошук" />
</td></tr></table>
</form>
</div>
<?
if (!empty($_POST)) {
$search = $_POST['string'];
}
if(isset ($_POST['search'])) {
$db = mysql_connect("localhost","root","");
$a=mysql_select_db("diplom", $db);
$search = substr($search, 0, 64);
$search = preg_replace("/[^\w\x7F-\xFF\s]/", " ", $search);
$good = trim(preg_replace("/\s(\S{1,2})\s/", " ", ereg_replace("
+", " ", $search )));
if($_POST['logic']=='OR'){
if ($a) {echo "";} else {echo "error!";}
$sss= "SELECT `url`, `nameproduct`, `price`, `description` FROM
`product` WHERE nameproduct LIKE '%" . str_replace(" ", "% ' OR
nameproduct LIKE '%" . $good). "% ' OR description LIKE '%" .
str_replace(" ", "% ' OR description LIKE '%" . $good). "%'";
$result = mysql_query ($sss,$db);
if ($result) {echo "";} else {echo "er-ror!";}
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {
$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){
$key=$z;
$z++;
if($key==1) {
```

```

echo"<table><tr><td rowspan=\"3\" width=\"120px\">";
if($buf!=NULL){
echo "<img src='$buf' alt='' class='leftimg'>";}
if($buf==NULL){
echo '';}
echo"</td><td>";}
if($key==2){
echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){
echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){
echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}
echo"</div>";}}
if($_POST['logic']=='AND'){
if ($a) {echo "";} else {echo "error!";}
$sss= "SELECT `url`, `nameproduct`, `price`, `description` FROM
`product` WHERE nameproduct LIKE '%" . str_replace(" ", "%") AND
nameproduct LIKE '%", $good). "%' OR description LIKE '%" .
str_replace(" ", "%") AND description LIKE '%" , $good). "%'";
$result = mysql_query ($sss,$db);
if ($result) {echo "";} else {echo "er-ror!";}
while ($myrow = mysql_fetch_array($result,MYSQL_ASSOC)) {
$z=1;
echo"<div id=\"product\">";
foreach ($myrow as $key => $buf){
$key=$z;
$z++;
if($key==1){
echo"<table><tr><td rowspan=\"3\" width=\"120px\">";
if($buf!=NULL){
echo "<img src='$buf' alt='' class='leftimg'>";}
if($buf==NULL){
echo '';}
echo"</td><td>";}
if($key==2){
echo "<p class=\"name\">$buf</p>";
echo"</td></tr><tr><td>";}
if($key==3){
echo "<p class=\"price\">$buf грн.</p>";
echo"</td></tr><tr><td>";}
if($key==4){
echo "<p class=\"description\">$buf</p>";
echo"</td></tr></table>";}}
echo"</div>";}}}?></div>

```

ДОДАТОК Л

/*stylez.css*/

```
html{
background-color:#bbbbbb;}
body{
position:relative;
padding-bottom:10px;
margin:0%;
width:100%;
background-color:white;
min-height:120%;}
h1{
text-align:center;}
ul{
list-style: none;
margin: 0;
padding-left: 0;
background:#737373;
height: 7%;}
ul a{
text-decoration: none;
background:#737373;
color:#fff;
padding:1%;
line-height:270%;
display: block;
border-left: 1px solid #ffffff;
border-right: 1px solid #ffffff;
-moz-transition: all 0.3s 0.01s ease;
-o-transition: all 0.3s 0.01s ease;
-webkit-transition: all 0.3s 0.01s ease;}
ul a:hover{
background:#333;}
li{
position:relative;
left:23%;
text-align:center;
width:18%;
display: inline-block;
font-size: 95%;}
.ul{
position:relative;
width:100%;
border-bottom:1px solid #ffffff;
border-top:1px solid #ffffff;}
.head{
position:relative;
height:110px;}
.text{
position:relative;
```

```

width:98%;
left:1%;
min-height:120%;
background:white;}
#login{
position:absolute;
width:100%;
height:100px;
padding-top:10;
padding-left:10;
top:0px;
background: black;
opacity: 0.7;
font-size:100%;}
#logina{
position:absolute;
height:103px;
width:100%;
top:0px;
padding-top:10;
padding-left:10;
background: black;
opacity: 0.7;}
.reg{
position:absolute;
left:37%;}
#selcat {
position:absolute;
z-index:1;
color: white;
font-weight:bold;
font-size: 100%;
left:-1.11%;
top:-23px;
overflow:hidden;
background:#737373;
width:23.7%;
display: inline-block;}
#selcat0{
color: white;
background:#fff;
text-decoration: none;
border:0px;}
#selcat option{
height:25px;
padding:10px;
-moz-transition: all 0.3s 0.01s ease;
-o-transition: all 0.3s 0.01s ease;
-webkit-transition: all 0.3s 0.01s ease;}
#selcat option:hover{
background:#333;}
#txt{

```



```

position:relative;
background:white;
width:76%;
min-height:120%;
left:24%;}
#product{
position:relative;
min-height:111px;
padding:1%;
margin-bottom:1%;
outline: 1px solid #000;}
.leftimg{
position:relative;
right:0%;
padding-right:1%;}
#product img {
height:200px;
width:200px;}
.imgnull {
position:relative;
float:left;
padding-right:1%;}
.name{
font-weight:bold;
font-size:120%;
text-align:center;}
.price{
font-weight:bold;
font-size:100%;
color:red;}
.description{
font-size:100%;
text-align:justify;}
#search{
position:absolute;
top:320px;
left:-32.5%;
width:23.7%;}
#sprice{
position:absolute;
top:433px;
left:-32.5%;}
footer{
position:absolute;
width:99%;
bottom:-67px;
color:white;
background:#737373;
padding:5px;
height:55px;
border-top: 1px solid #000;
text-align:right;}

```

```
footer a{
margin-right:-5px;
padding:5px;
color:white;
text-decoration:none;
-moz-transition: all 0.3s 0.01s ease;
-o-transition: all 0.3s 0.01s ease;
-webkit-transition: all 0.3s 0.01s ease;}
footer a:hover{
background:#333;}
#seldel{
position:relative;
float:right;
right:0;}
#seldel select{
position:absolute;
overflow:hidden;
bottom:1px;
width:40px;
right:0;}
#seldel option{
text-align:center;
font-weight:bold;
height:15px;
padding-top:5px;}
#seldel option:hover{
background:red;
color:white;}
#change{
position:relative;
right:0;
float:right;}
h2{position:absolute;right:100px;}
```