КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій
(факультет)

інформаційних технологій проєктування та прикладної математики
(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

на тему: «Інтегрована узагальнена система автоматизації розробки вебсайту комп'ютерної техніки»

ЯКОВЛЄВ ІЛЛЯ ЛЕОНІДОВИЧ
(прізвище, ім’я та по батькові студента повністю)

Київ 2025 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проєктування та прикладної математики

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.

„___” _____2025 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

на тему: " Інтегрована узагальнена система автоматизації розробки вебсайту комп'ютерної техніки "

Виконав: студент ІІ-го курсу, групи ІСТм-24

Спеціальності: 126 «Інформаційні системи та технології»

Яковлєв І.Л.
(прізвище та ініціали)

Керівник к.т.н., проф. Теренчук С.А.
(прізвище та ініціали)

Рецензент к.т.н., доц. Шабала Є.Є.
(прізвище та ініціали)

Київ, 2025 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: автоматизації і інформаційних технологій
Кафедра: інформаційних технологій проєктування та ПМ
Освітній рівень: “магістр за ОПП”
Спеціальність: 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.
_____
“___” _____2025 року

З А В Д А Н Н Я
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

Яковлєв Ілля Леонідович

1. Тема роботи: «Інтегрована узагальнена система автоматизації розробки вебсайту комп'ютерної техніки»
затверджена наказом ректора КНУБА № 1619/23/25 від 29.09.2025 р.
2. Керівник роботи: к.т.н., професор Теренчук Світлана Анатоліївна, професор кафедри інформаційних технологій проєктування та ПМ
3. Строк подання студентом роботи до захисту: грудень 2025 року
4. Зміст пояснювальної записки за розділами:
Р.1. Аналіз предметної області
Р.2. Проектування структури та бази даних
Р.3. Реалізація інтерфейсу Web-сайту
Р.4. Ергономіка ІТ
5. Інформаційні слайди:
С.1. Функціонуюча структура Інтернет-магазину
С.2. Структура бази даних
С.3. Структура Web-сайту Інтернет-магазину комп’ютерної техніки

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р.1. Аналіз предметної області.	Вересень 2025 р.
Р.2. Проектування структури та бази даних.	Жовтень 2025 р.
Р.3. Реалізація інтерфейсу Web-сайту.	Листопад 2025 р.
Р.4. Ергономіка ІТ	Грудень 2025 р.
Остаточне оформлення роботи	Грудень 2025 р.
Направлення роботи на рецензування, перевірку на плагіат	Грудень 2025 р.
Попередній захист роботи на кафедрі	Грудень 2025 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Прийом програмного продукту	к.т.н., доц. Шабала Є.Є.		

8. Дата видачі завдання: 22 вересня 2025 року

Керівник _____ Теренчук С.А.
(підпис) (прізвище та ініціали)

Магістрант _____ Яковлєв І.Л.
(підпис) (прізвище та ініціали)

РЕЗЮМЕ

Київський національний університет будівництва і архітектури

Яковлєв Ілля Леонідович

факультет автоматизації і інформаційних технологій,

група ІСТм-ІІ

Тема кваліфікаційної роботи:

Інтегрована узагальнена система автоматизації розробки вебсайту комп'ютерної техніки

освітній рівень: магістр,

спеціальність: 126 «Інформаційні системи і технології»,

Науковий керівник: Теренчук Світлана Анатоліївна ,

кандидат технічних наук, професор кафедри інформаційних технологій проектування та прикладної математики

Обсяг роботи. Кваліфікаційна робота магістра складається: розділів 5, стор. 75, таблиць 3, рис. 42, завдання, анотація, вступу, висновків, списку використаних джерел.

Актуальність даної теми. Метою даної роботи є показати застосування технологій PHP і MySQL при створенні Web – сайту комп’ютерної техніки.

Актуальність даної теми: обумовлена використанням веб-технологій, які помітно зростають в усіх сферах людського життя. На сьогоднішній день технології PHP і MySQL отримують широкий розвиток, також на даний час актуальні веб - ресурси, які служать в якості інформативних сайтів.

В першому розділі проводиться аналіз предметної області та постановка задачі, поняття Веб-сайт, класифікація Веб – сайтів, технологія PHP, технологія MySQL.

В другому розділі проводиться проектування БД

У третьому розділі проведено реалізація інтерфейсу Web-сайту.

У четвертому розділі «Ергономіка інформаційних технологій».

Досліджено аналіз небезпечних та шкідливих виробничих факторів Запропоновано програмно-технічний комплекс практичної реалізації системи;

розраховано час евакуації людей при пожежі в приміщенні . В розділі на реальних прикладах описано ергономічні вимоги ***до організації обладнання робочих місць з комп'ютерною технікою.***

Ключові слова: технології, застосування, база даних, веб-сайт.

Keywords: technologies, application, database, website.

Якість оформлення проекту. Кваліфікаційна робота магістра оформлена у відповідності до діючих нормативних документів та методичних вказівок до виконання дипломної роботи для студентів спеціальності 126. «Інформаційні системи і технології».

Загальний висновок стосовно роботи та присвоєння авторові освітнього рівня «магістр». Робота виконана на високому рівні, студент продемонстрував високий рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслуговує оцінки «добре».

Науковий керівник _____/ проф. Теренчук С.А. /

(підпис)

Посада, місце роботи. КНУБА, пр-т. Повітряних Сил, 31, професор кафедри інформаційних технологій проектування та прикладної математики

«8» грудня 2025 р.

АНОТАЦІЯ

«Інтегрована узагальнена система автоматизації розробки вебсайту комп'ютерної техніки».

Кваліфікаційна робота магістра за спеціальністю: 126. «Інформаційні системи і технології» – Київський національний університет будівництва та архітектури. – Київ, 2025.

Метою даної роботи є показати застосування технологій PHP і MySQL при створенні Web – сайту комп’ютерної техніки.

Актуальність даної теми: обумовлена використанням веб-технологій, які помітно зростають в усіх сферах людського життя. На сьогоднішній день технології PHP і MySQL отримують широкий розвиток, також на даний час актуальні веб - ресурси, які служать в якості інформативних сайтів.

Ключові слова: технології, застосування, база даних, веб-сайт.

SUMMARY

"Integrated generalized automation system for computer technology website development."

Master's attestation master's degree in specialty: 126. "Information systems and technologies". - Kyiv National University of Civil Engineering and Architecture - Kiev, 2025.

The purpose of this work is to use PHP and MySQL technologies when creating a website that uses technical techniques.

Relevance of this topic: due to the use of web technologies, which are growing significantly in all spheres of human life. Today, PHP and MySQL technologies are widely developed, as well as currently relevant web resources that serve as informative sites.

Keywords: technologies, application, database, website.

ЗМІСТ

ВСТУП 9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 11
1.1 Поняття Веб-сайт 11
1.2 Класифікація Веб – сайтів 11
1.3 Локальний сервер OpenServer 14
1.4 HTML 15
1.5 CSS 17
1.6 Технологія PHP 19
1.7 Технологія MySQL 34
1.8 Веб-додаток phpMyAdmin 39
2. ПРОЕКТУВАННЯ СТРУКТУРИ ТА БАЗИ ДАНИХ 43
2.1 Організація діяльності Web-сайту 43
3. РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ WEB-САЙТУ 49
4. ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ……56
ВИСНОВКИ 68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 69

ВСТУП

У дипломній роботі розглянуто технології PHP і MySQL при створенні Web – сайту комп’ютерної техніки.

З появою веб-технологій комп'ютер починають використовувати абсолютно нові верстви населення. Можна виділити дві найбільш характерні групи, що знаходяться на різних соціальних полюсах, які були стрімко залучені в нову технологію, можливо, навіть крім їх власного бажання. З одного боку, це були представники елітарних груп суспільства – керівники крупних організацій, президенти банків, топ-менеджери, впливові державні чиновники і так далі. З іншого боку, це були представники найширших верств населення – домогосподарки, пенсіонери, діти.

При появі технології веб, комп'ютери обернулися лицем до цих двох абсолютно протилежним категорій потенційних користувачів. Еліту об'єднувала одна межа – через високу відповідальність і практично стовідсотковій зайнятості “великі люди” ніколи не користувалися комп'ютером; типовою була ситуація, коли з комп'ютером працював секретар. У якийсь момент часу вони зрозуміли, що комп'ютер їм може бути корисний, що вони можуть результативно використовувати той невеликий час, який можна виділити на роботу за комп'ютером. Вони раптом зрозуміли, що комп'ютер – це не просто модна і дорога іграшка, але інструмент отримання актуальної інформації для бізнесу. При цьому їм не потрібно було витрачати багато часу, щоб освоїти технологію роботи з комп'ютером (в порівнянні з тим, як це було раніше).

Спектр соціальних груп, що підключаються до мережі Інтернет і що шукають інформацію в WWW, весь час розширюється за рахунок користувачів, що не відносяться до категорії фахівців в області інформаційних технологій. Це лікарі, будівельники, історики, юристи, фінансисти, спортсмени, мандрівники, священослужителі, артисти, письменники, художники. Список можна продовжувати бескінченно. Будь-який, хто відчув корисність і незамінність мережі для своєї професійної діяльності або захоплень, приєднується до

величезної армії споживачів інформації в «Всесвітній Павутині».

Веб-технологія повністю перевернула наші уявлення про роботу з інформацією, та і з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна спроможність, ємкість пристроїв, що запам'ятовують, не враховували головного «вузького місця» системи – інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І лише коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, послідував безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

З розвитком технологій гіпертекстової розмітки в Інтернеті почали з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів крупних компаній, що оповідають про успіхи компанії і її провали, до сайтів маленьких фірм, що пропонують відвідати їх офіси в межах одного міста.

Розвиток Інтернет-технологій послужив поштовхом до появи нової вітки в Інтернеті – Інтернет-форумів. Почали з'являтися сайти, і навіть цілі портали, на яких люди зі всіх куточків планети можуть спілкуватися, отримувати відповіді на будь-які питання і, навіть, укладати ділові операції.

Також в наш час стає поширеним перегляд будь-яких товарів в інтернеті.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття Веб-сайт

Веб-сайт – це одна або сукупність веб-сторінок, доступних в Інтернеті через протоколи HTTP/HTTPS. Веб-сайт - це місце в Інтернеті, яке визначається своєю адресою (URL), має свого власника і складається з веб-сторінок, які сприймаються як єдине ціле.

Сукупність всіх загальнодоступних веб-сайтів є Всесвітня павутина. Сторінки веб-сайту об'єднані загальною кореневою адресою, а також звичайно темою, логічною структурою, оформленням або авторством. Сторінки веб-сайтів – це файли з текстом, розміченим на мові HTML або XHTML. У HTML або XHTML -коді сторінки містять лише вказівки на такі файли.

Ці файли, будучи завантаженими відвідувачем на його комп'ютер, обробляються програмою-оглядачем, званою браузером і виводяться на засіб відображення користувача (монітор, екран КПК, принтер або синтезатор мови). Мова HTML/XHTML дозволяє форматувати текст, розрізняти в ньому функціональні елементи, створювати гіпертекстові посилання (гіперпосилання) і вставляти в сторінку, що відображається, зображення, звукозаписи і інші мультимедійні елементи. Відображення сторінки можна змінити додаванням в неї таблиці стилів на мові CSS або сценаріїв на мові JavaScript.

Сторінки сайтів можуть бути простими статичними наборами файлів або створюватися спеціальною комп'ютерною програмою на сервері - так званим движком сайту. Движок може бути або зроблений на замовлення для окремого сайту, або готовим продуктом, розрахованим на якийсь клас сайтів. Деякі з движків можуть забезпечити власнику сайту можливість гнучкої настройки структуризації і виведення інформації на веб-сайті; такі називаються системами управління змістом.

1.2 Класифікація Веб – сайтів

За доступністю сервісів:

- Відкриті сайти. Всі сервіси сайту цілком є доступними для всіх відвідувачів і користувачів.

- Напіввідкриті сайти. Для доступу до ресурсів чи сервісів сайту необхідно зареєструватися, зазвичай, безкоштовно.

- Закриті сайти. Цілком закриті службові сайти організацій, зокрема, корпоративні сайти, особисті сайти приватних осіб. Вони є доступними для вузького кола користувачів. Доступ для нових користувачів, зазвичай, надається через запрошення (invite) або через адміністратором до переліку допущених осіб.

За природою вмісту:

- Статичні сайти. Вміст сайту формується розробником і в подальшому не змінюється. Користувач бачить файли в тому вигляді, в якому вони зберігаються на сервері.

- Динамічні сайти. Вміст генерується спеціальними програмами (скриптами) і формується залежно від дій користувача на підставі інших даних з визначених джерел.

За фізичним розташуванням:

- Зовнішні сайти мережі Інтернет.

- Локальні сайти, які є доступними лише в межах локальної мережі. Це можуть бути як корпоративні сайти організацій, так і сайти приватних осіб в локальній мережі провайдера.

За схемою представлення інформації, її об'єму і категорії вирішуваних завдань:

- Інформаційні ресурси:

- Тематичний сайт – сайт, що надає специфічну інформацію з певної теми.

- Тематичний портал – надвеликий ресурс, який надає вичерпну інформацію з певної тематики. Портали є подібними до тематичних сайтів, але додатково містять засоби взаємодії з користувачами і дозволяють користувачам спілкуватися в рамках порталу (форуми, чати).

- Інтернет представництва власників бізнесу (торгівля чи послуги, які не обов’язково безпосередньо пов'язані з Інтернетом):

– Сайт-візитка – містить загальні дані про власника сайту (організацію або приватну особу): сфера інтересів, біографія, фотогалерея, контактні дані. Тобто, це є докладна візитна картка.

– Представницький сайт – так іноді називають сайт-візитку з розширеною функціональністю: докладний опис послуг, прас-листи, реквізити, схема проїзду, відгуки, форма зворотного зв'язку тощо.

– Корпоративний сайт – містить повну інформацію про компанію, послуги, продукцію, події в житті компанії. Відрізняється від сайту-візитки і представницького сайту повнотою представленої інформації, часто містить різні функціональні інструменти для роботи з контентом (пошук інформації, календарі подій, фотогалереї, корпоративні блоги, форуми). Може бути інтегрованим з внутрішніми інформаційними системами компанії (наприклад, бухгалтерськими системами). Може містити закриті розділи для різних груп користувачів – співробітників, дилерів, контрагентів.

– Каталог продукції – в каталозі надано докладний опис товарів/послуг, сертифікати, технічні і споживчі дані, відгуки експертів тощо. На таких сайтах розміщується розширена інформація про товари чи послуги, яку зазвичай не вказують у прайс-листі.

– Промо-сайт – сайт про конкретну торгову марку або продукт, на таких сайтах розміщується вичерпна інформація про бренд, різні рекламні акції (конкурси, вікторини, ігри тощо).

– Інтернет-магазин – сайт з каталогом продукції, за допомогою якого клієнт може замовити потрібні йому товари. Використовуються різні системи розрахунків: пересилка товарів післяплатою, автоматична пересилка рахунку факсом, розрахунки за допомогою пластикових карт чи інші типи розрахунків.

- Веб-сервіс — це послуга, що створена для виконання певних завдань в рамках служби Веб:

– Дошка оголошень.

– Каталог сайтів.

– Пошукові системи.

– Поштовий сервіс.

– Веб-форуми.

– Блоговий сервіс.

– Сервіс зберігання та обміну файлів.

– Сервіс створення та редагування документів – наприклад, Google Docs.

– Зберігання фотографій – наприклад, Picnik, ImageShack, Panoramio, Photobucket.

– Зберігання відео – наприклад, YouTube, Dailymotion.

– Соціальні Медіа – наприклад, Buzz.

– Соціальні мережі – наприклад, Facebook.

– Спеціалізовані соціальні мережі – наприклад, Odnoklassniki.

1.3 Локальний сервер OpenServer

Для розробки Web-сайту на персональному комп'ютері необхідно локальний сервер.

Для операційної системи Windows використаємо сервер: OpenServer.

Система OpenServer проста в установці та видаленні, з нею досить просто працювати завдяки зручному інтерфейсу. OpenServer включає пакет веб-серверів Nginx, який базується на аналозі Apache. Nginx працює з завданнями, що надходять від користувача, і публікує статичний вміст. У свою чергу Apache публікує динамічний вміст.

Nginx використовується на сторінках таких соціальних мереж, як:

Facebook. Найкраще, що Nginx та Apache демонструють себе одночасно.

OpenServer включає кілька модулів HTTP, три СУБД і велику кількість модулів PHP. Однак це перемикання між модулями виконується без будь-яких труднощів у оператора.

OpenServer швидкий, не дивлячись на великі розміри. Для установки системи потрібен великий обсяг пам'яті. Немає відмінностей від попередників система не "засмічує" системні файли. Також систему можна встановити на пристрій, придатний для розробника.

1.4 HTML

HTML (від англ. Hypertext Markup Language - мова розмітки гіпертексту) - це стандартна мова розмітки документів у Всесвітній павутині. Всі веб-сторінки створюються за допомогою мови HTML (або XHTML). Мова HTML інтерпретується браузером і відображається у вигляді документа, зручного для людини. HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879.

HTML-документ є текстовим файлом, розміченим за допомогою спеціальних (текстових) команд. Текстовий формат представлення веб-документів був вибраний, виходячи з основних вимог до веб-документу: простота, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу, зручність редагування і інтерпретації.

Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів (у оригіналі element), що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за допомогою команд розмітки, званих тегами (відангліського tag - ярлик). Всі HTML-теги, що зустрічаються в тексті документа, інтерпретуються браузером при відображенні документа.

З 1994 року по теперішній час розробка проводиться під егідою наддержавної організації World Wide Web Consortium. Інформацію "з перших рук" про стандарти, рекомендації і перспективи розвитку не тільки мови HTML,

але і цілого ряду інших web-технологій, можна знайти в Інтернеті за адресою http://www.w3.org. Вся документація на сайті W3C представлена на англійській мові, проте є і посилання на переклади (зокрема, на російську мову).

Основними віхами на шляху розвитку стандартів HTML можна вважати ухвалення стандартів HTML 1.2, HTML 2.0, HTML 3.2, HTML 4.0. На момент написання цієї статті останньою розробкою є стандарт HTML4.01.

Версії:

- RFC 1866 – HTML 2.0, схвалений як стандарт 22 вересня 1995;
- HTML 3.2 – 14 січня 1996;
- HTML 4.0 – 18 грудня 1997;
- HTML 4.01 (незначні зміни) – 24 грудня 1999;
- ISO/IEC 15445:2000 (так званий ISO HTML, заснований на HTML 4.01 Strict) – 15 травня 2000 [5].

До створення Web-сторінок існує три загальних підходи:

Високий рівень – створення документу звичайними засобами з наступною його конвертацією в HTML,

Середній рівень – створення Web-сторінки в спеціалізованому HTML-редакторі,

Низький рівень – створення Web-сторінки безпосередньо в HTML-кодах.

Перший підхід не вимагає жодних знань мови HTML. Документ створюється в довільному текстовому редакторі у форматі, в якому він буде відображатись у Web-браузері. Після створення документ конвертується уHTML-код. Наприклад, Microsoft Word дозволяє зберегти відкритий документ у форматі HTML. Але цей підхід має суттєві недоліки: по-перше, код, що отримано в результаті конвертації, є досить громіздким і незручним для подальшого редагування, а по-друге, можливості Web-сторінки, що отримана таким шляхом, досить обмежені, оскільки від початку створюється не HTML-документ, а, наприклад, Word-документ.

Другий підхід є найзручнішим з точки зору співвідношення часу, що витрачається на створення Web-сторінки, та можливостей цієї сторінки, отриманих за допомогою спеціалізованого HTML-редактора. Такі HTML-

редактори, як правило, дозволяють створювати Web-сторінки подібно тому, як створюються документи у текстових процесорах, із форматуванням тексту, графіки, таблиць та інших елементів сторінки за принципом WYSIWYG (What You See Is What You Get – "що бачиш, те і отримаєш"), при цьому сам документ зберігається відразу у HTML-коді. Крім того, такі HTML-редактори дозволяють безпосередньо редагувати HTML-код. Ефективність отриманого коду при цьому підході залежить від ефективності та можливостей вибраного редактора.

Третій підхід вимагає глибокого знання мови HTML. Редагування Web-сторінок відбувається у форматі, дещо незручному для сприйняття початківцем. Однак при такому підході можна створити дуже ефективний HTML-код, що забезпечить максимальну інформативність сторінки при її мінімальному обсягу. Крім того, в такому випадку Web-дизайнеру доступні всі можливості мови HTML, які можуть не бути реалізовані у спеціалізованих HTML-редакторах.

1.5 CSS

CSS - це мова каскадних таблиць стилів (англ. cascading style sheets) , яка використовується для опису стилів багаторазового використання для подання документів, написаних мовою розмітки. Його концепція була створена компанією Håkon Wium Lie у 1994 році. У грудні 1996 року WSC розробила специфікацію для CSS і сьогодні дозволяє веб-розробникам змінювати макет і зовнішній вигляд своїх веб-сторінок. Наприклад, CSS може використовуватися для зміни шрифту, який використовується в певному HTML-елементі, а також його розміру та кольору. Один CSS-файл може бути пов'язаний з декількома сторінками, що дозволяє розробнику одночасно змінювати зовнішній вигляд усіх сторінок. Наведене нижче поле містить основний приклад використання коду CSS для визначення шрифтів, кольору гіперпосилань та кольору посилання, коли курсор миші наводить курсор миші. У цьому конкретному прикладі ми змінюємо лише теги HTML <a> та <body>, а не створюємо нові селектори класу чи id.

```
body {
 font: normal 100% "trebuchet ms", Arial, Helvetica, sans-serif;
}
a {
 color: #000000;
}
A:visited {
 color: #005177;
}
a:hover {
 color: #005177;
}
```

Рис.1.1. Поле з CSS кодом

Переваги CSS:

1. CSS економить час - ви можете написати CSS один раз, а потім повторно використовувати той же аркуш на декількох HTML-сторінках. Ви можете визначити стиль для кожного елемента HTML і застосувати його до якомога більше веб-сторінок.

2. Сторінки завантажуються швидше - якщо ви використовуєте CSS, вам не потрібно щоразу писати атрибути тегів HTML. Просто напишіть одне правило CSS тегу та застосуйте його до всіх випадків цього тегу. Так менше коду означає швидше завантаження.

3. Простота обслуговування - Щоб зробити глобальну зміну, просто змініть стиль, і всі елементи на всіх веб-сторінках будуть оновлені автоматично.

4. Покращені стилі до HTML - CSS має набагато ширший набір атрибутів, ніж HTML, тому ви можете надати набагато кращий погляд на свою сторінку HTML порівняно з атрибутами HTML.

5. Сумісність декількох пристроїв - аркуші стилів дозволяють оптимізувати вміст для більш ніж одного типу пристроїв.

Використовуючи один і той же документ HTML, різні версії веб-сайту можуть бути представлені для портативних пристроїв, таких як КПК та мобільних телефонів або для друку.

Версії CSS:

Каскадні таблиці стилів рівня 1 (CSS1) вийшли з W3C як рекомендація в грудні 1996 року. Ця версія описує мову CSS, а також просту модель візуального форматування для всіх тегів HTML.

CSS2 став рекомендацією W3C у травні 1998 року та базується на CSS1. Ця версія додає підтримку для специфічних таблиць стилів, наприклад, принтери та аудіопристрої, завантажувані шрифти, розміщення елементів та таблиці.

1.6 Технологія PHP

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

PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта.

На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript- кодів які виконаються вже на стороні клієнта.

Особливості

PHP — мова, яка може бути вбудована безпосередньо в html- код сторінок, які, в свою чергу коректно будуть оброблені PHP - інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустріне парну екрануючу послідовність (?>).

Велика різноманітність функцій PHP дають можливість уникнути написання багаторядкових призначених для користувача функцій на C або

Pascal.

– Наявність інтерфейсів до багатьох баз даних

– в PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.

– через стандарт відкритого інтерфейсу зв'язку з базами даних (Open Database Connectivity Standard — ODBC) можна підключатися до всіх баз даних, до яких існує драйвер.

– Традиційність

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl і C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що на даний момент є мало не найпопулярнішою мовою для створення веб-застосунків (скриптів).

– Наявність вихідного коду та безкоштовність

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили благотворний вплив на багато проектів, в першу чергу — Linux хоч і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту PHP.

Ухвалення стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціниму послугу користувачам. Додатково, користувачі PHP в усьому світі є свого роду колективною службою підтримки, і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

– Ефективність

Ефективність є дуже важливим чинником при програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте, щоб не робили розробники PHP, виконувані файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP цілком достатня для створення більш-менш серйозних веб-застосунків.

З точки зору типізації, PHP є мовою програмування з динамічною типізацією. Немає необхідності явного визначення типу змінних, хоча така можливість існує. В разі звернення до змінної, ядро PHP трактує її тип відповідно до контексту. За необхідності можливе приведення змінної до певного типу за допомогою відповідних конструкцій мови. Це може знадобитись, якщо зважити, що значення змінної можуть трактуватись по-різному в залежності від її типу. Також можливе визначення типу відповідної змінної на певному етапі виконання сценарію. Імена змінних чутливі до регістру символів.

До базових типів належать булеві дані, цілі та дійсні числа із плаваючою комою, а також строки. Булеві дані виражають істинність значення. Цілі числа можуть бути подані у вісімковому, десятковому та шістнадцятковому вигляді. Розмір цілого числа може змінюватись залежно від платформи, як правило, розрядність становить 32 біти. PHP не підтримує беззнакові цілі числа. Дійсні числа із плаваючою комою можуть бути подані в десятковій або експоненційній формі. Строки розділяють на два класи – строки, що підлягають аналізу, та строки, що не підлягають аналізу. Перший клас досліджується інтерпретатором на наявність посилань на інші змінні і за умови їх наявності, відбувається підстановка значень у відповідне місце. Крім того, клас дозволяє проводити маніпуляції із керуючими символами. Символ строки може мати

лише одне із 256 значень, але існує можливість працювати із багатобайтовими символами. Доступ до символів строки можливий з використанням синтаксису, схожого на доступ до елементів масивів.

PHP надає широкий спектр функцій для пошуку та заміни тексту в строках. Для цього використовують як традиційний підхід, так і спеціальний підхід, що базується на використанні регулярних виразів. При цьому в мові реалізована підтримка двох видів регулярних виразів – Perl-сумісні та POSIX-сумісні, що розрізняються за синтаксисом та особливостями роботи.

До змішаних типів належать масиви, хеші та об’єкти. Масиви в сенсі мови є наборами змінних, що згруповані в єдину змінну. Вимога однотипності наповнення масивів не ставиться. Технічно, масиви являють собою впорядковані карти, що відображають ключові значення на позиції змінних даних. Вмістом значення, на яке вказує ключ, може бути будь-чим, що можна подати у вигляді змінної. Не існує жодних обмежень, крім об’єму пам’яті, що накладаються на кількість ключів масиву.

Особливістю мови є відмова від рівномірного розподілу ключів масивів. Знайшовши реалізацію і модель багатовимірних масивів, причому без явного обмеження глибини вкладеності. Корисною властивістю PHP є можливість асоціації масивів із функцією зворотного виклику. Ці функції дозволяють проводити дії над одним чи кількома масивами в пакетному режимі.

Також існують два спеціальні типи даних – ресурс та NULL. Ресурс – спеціальна змінна, що містить посилання на зовнішній ресурс. Ресурси створюються та використовуються в спеціалізованих функціях. Оскільки тип містить спеціальні вказівники на відкриті файли, приєднання та інше, то будь-які дії над значенням ресурсу не мають сенсу.

Область видимості змінної – середовище, в якому вона визначена. Розрізняють локальні та глобальні змінні. За замовчуванням, всі змінні мають локальний характер дії. Виділяють особливий тип змінних – статичні, що дозволяє повторне звернення до змінної в певному сегменті коду, причому змінна буде зберігати попередньо отримане значення. Існує також поняття суперглобальних змінних, які служать місцем збереження данихоточення або

даних, отриманих ззовні. Підтримується концепція динамічних змінних та функцій.

Константи в PHP – ідентифікатори простих значень. Можливе визначення константи, причому після її оголошення стає неможливою зміна її значення чи анулювання. Константи можуть мати лише скалярні значення. Підтримується можливість отримання значення константи за динамічним ім'ям. Область видимості констант буде глобальною для сценарію та всіх під'єднаних компонентів. Також в ядрі мови визначено чимало системних констант.

Оператори в сенсі мови дозволяють виконувати відповідну дію над одним чи кількома операндами. Оператори бувають трьох типів – унарні, бінарні та тернарні. Оператори, як і в інших мовах, характеризуються не лише дією, а й асоціативністю та пріоритетністю. Особливістю булевих операцій порівняння – розрізнення двох класів – з врахуванням типу і без врахування типу, при якому відбувається приведення до відповідного типу. Заокруглення відбуваються завжди в меншу сторону. В мові реалізовані особливі класи операторів – виконання, управління помилками та перевірки приналежності до класу.

Функції в сенсі мови є контейнерами коду, причому можливе поміщення інших функцій та класів. На цьому і базується можливість умовного визначення функції. В цьому випадку висувається вимога попередньої декларації викликаної функції, що не обов'язково в інших випадках. Можливості перевизначення чи деактивації функції не існує. Результат, який повертає функція, може мати будь-який тип.

В мові реалізована функціональність посилань. Можливо створити скільки завгодно псевдонімів, що посилаються на єдиний сегмент даних. При вивільненні будь-якого з псевдонімів, сегмент даних залишається в пам'яті до моменту завершення сценарію або вивільнення усіх посилань.

Що стосується функцій в PHP, то замість прийнятого в багатьох мовах принципу перевантаження функцій, що дозволяє змінити хід виконання функції в залежності від типу та кількості переданих параметрів, використовується метод динамічних аргументів. Це дає змогу не визначати кількість параметрів

для функцій при їх оголошенні, а працювати із тими аргументами, які були отримані на момент виклику функції. У тілі функції можливо отримати кількість переданих їй аргументів і проводити відповідні маніпуляції. При оголошенні функції звичайним чином, можливе завдання значень аргументів за замовчуванням. Функції можуть повертати лише одне значення, проте це обмеження можна оминути, використавши не лише масиви, а й посилання. Передача аргументів за посиланням неможлива під час виконання та оголошення функції.

Після виконання сценаріїв, простір пам'яті, займаної ними, очищується збирачем сміття. Проте, за необхідності, можливе виконання очищення пам'яті від надлишкових сегментів даних під час виконання скриптів. Використання функцій очищення пам'яті є невиправданим, хоча така можливість існує. Для побудови програмних комплексів можливо використовувати модульний підхід, виконуючи розділення різнорідного коду. При потребі, можливе виконання приєднання необхідних модулів, причому операція виконання може бути і умовною. Приєднані до скрипта файли можуть повертати значення. Мова явно підтримує HTTP cookies відповідно до специфікацій Netscape. Це дозволяє проводити встановлення та читання невеликих сегментів даних на стороні клієнта.

Протокол HTTP, засобами якого, як правило, обмінюються інформацією клієнт та Web-сервер, не надає змогу зберегти стан сеансу взаємодії. Це випливає із того, що між клієнтом та сервером не встановлюється постійне з'єднання і клієнт не надає жодних відомостей, що можуть виділити його з поміж інших активних в деякому околі часу.

Альтернативою cookies є концепція сесій, яка знайшла свою реалізацію в PHP. В сесії можна зберігати різні дані, включаючи об'єкти.

Історія

Перші версії

Історія PHP починається з 1995 року, коли Расмус Лердорф (Rasmus Lerdorf) створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на веб-сайті. Потім, коли цим застосунком вже

користувалися кілька чоловік, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Personal Home Page Tools версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.

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

У тому ж 1997 році до проекту PHP підключилися Зеєв Сураський (Zeev Suraski) і Енді Гутманс (Andi Gutmans). Ці студенти Техніону, одного з ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зеєв і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягомдекількох місяців вони блискуче впоралися з цим завданням.

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

При цьому головною особливість PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість

працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 кількість користувачів PHP перевалила за 100000, а PHP був вже встановлений на не менше ніж 10% серверах Інтернету. У той ж час значному поширенню даної мови сприяли публікації в електронній пресі та вихід книжок для вивчення PHP.

Відразу ж після виходу PHP 3, Енді Гутманс і Зеєв Сураський почали переробку ядра PHP. В першу чергу належало вирішити проблему підвищення продуктивності. Новий продукт, названий Zend Engine (від імен творців: Zeev і Andi), успішно справлявся з поставленим завданням і був реалізований в 1999 році. Основними реалізованними ідеями є можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.

Визнання та поширення

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

За час з 2000 по 2004 рік продовжувалися активні роботи з покращення 4 версії, але майже відразу PHP Group приступила до продумування можливостей нової версії. В першу чергу було вирішено підсилити об'єктні можливості мови, що дозволяло використовувати його для реалізації масштабних проектів. Роботи із створення версії 5 велися тривалий час, в них брало участь рекордна кількість фахівців, зокрема Стерлінг Хьюз (Sterling Hughes) і Маркус Бергера (Marcus Boerger).

У липні 2004 року виходить офіційний реліз PHP 5. В першу чергу, як і планувалося, було перероблено весь механізм роботи з об'єктами. І якщо в попередніх версіях об'єктно-орієнтоване програмування на PHP було можливе в мінімальному ступені, а томуі використовувалося на практиці не часто, то PHP 5 володіє прекрасним потенціалом реалізації об'єктного

програмування. Окрім цього, PHP збагатився рядом цінних розширень для роботи з XML, різними джерелами даних, генерації графіки і інше.

Серед інших украй корисних доповнень в PHP 5 слід зазначити нову схему обробки виключень. Конструкція try/catch/throw дозволяє весь код обробки помилок локалізувати водному місці сценарію.

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

У PHP 5 також включені два нові модулі для роботи з протоколами — SimpleXML і SOAP. SimpleXML дозволяє значно спростити роботу з XML-даними, представляючи вміст XML- документа у вигляді PHP-об'єкта. Розширення SOAP дозволяє будувати на PHP сценарії, що обмінюються інформацією з іншими застосунками за допомогою XML-повідомлень поверх існуючих веб- протоколів, наприклад HTTP. Модуль для роботи з SOAP для PHP 5 надає розробникам засіб для достатньо швидкого створення ефективних SOAP-клієнтів і SOAP-серверів.

Новий модуль PHP 5 MySQLi (MySQL Improved) призначений для роботи з MySQL-сервером версій 4.1.2 і вище, реалізовуючи не тільки процедурний, але і об'єктно-орієнтований інтерфейс до MySQL. Додаткові можливості цього модуля включають — SSL, контроль транзакцій, підтримка реплікації і ін.

PHP 6 та Unicode - PHP отримав змішані відгуки через відсутність власної підтримки Unicode на рівні основної мови. У 2005 році був започаткований проєкт, очолюваний Андрієм Змієвським, для залучення рідної підтримки Unicode на PHP, шляхом вбудовування бібліотеки «Міжнародні компоненти для Unicode» (ICU) та вбудованих текстових рядків як UTF-16. Оскільки це призведе до серйозних змін як до внутрішньої частини мови, так і до коду користувача, планувалося випустити його як версію 6.0 мови разом з іншими основними функціями, які розвиваються.

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

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

Синтаксис

Відокремлення PHP коду від HTML розмітки

Всі сценарії оформляються у вигляді блоків коду. Ці блоки можуть бути поміщені в HTML-код, але відділені від нього відповідними обмежувачами. Код PHP в HTML повинен знаходитись між початковим тегом *<?php* та кінцевим *?>* (або між *<script language="php">* та *</script>)* Бажаним варіантом виділення PHP коду є варіант *<?php ?>*, оскільки саме такі початковий та кінцевий теги дозволять використовувати PHP код в документах, які відповідають правилам XML. Також можна користуватися скороченим записом: *<? ?>* (інколи потрібно активізувати даний стиль внісши вручну зміни в файл php.ini: змінна *short_open_tag* повинна мати значення *On*) і записом в стилі ASP: *<% %>* (в php.ini змінна *asp_tags* повинна мати значення *On*). Проте стиль ASP не рекомендується і очікується, що він буде відсутній у PHP6.

Найростіша програма Hello world на PHP виглядає так:

```
<?php
  echo 'Hello, world!';
?>
```

PHP виконує код, що знаходиться в середині обмежувачів, таких як <?php ?>. Все, що знаходиться поза межами обмежувачів виводиться без змін. Таким чином виконується вставка PHP коду в HTML код. Наприклад, код html-сторінки з попереднім прикладом виглядатиме так:

```
<html>
 <head>
  <title>Тестуємо PHP</title>
 </head>
 <body>
  <?php echo 'Hello, world!'; ?>
 </body>
</html>
```

Відокремлення інструкцій

Інструкції в PHP відокремлюються символом **;**. Перед закінченням скрипту (перед тегом **?>**) крапку з комою ставити необов'язково.

Коментарі

Коментарі в PHP можуть бути як однорядкові так і багаторядкові. Однорядкові коментарі починаються із // або # (коментар скриптових мов UNIX). Багаторядкові коментарі починаються символами **/*** і закінчуються ***/** . Слід зазначити, що однорядкові коментарі йдуть або до кінця поточного рядка, або до завершального тега ?>.

Змінні

Основою будь-якого програмування є змінні. PHP, як і деякі інші UNIX-скриптові мови, не потребує їх опису. Існує три стилі оформлення змінних у PHP:

– *короткий стиль.* Змінні записуються у вигляді: *$variable.* Використовується в більшості випадків. Змінна, яка створюється програмним

кодом, записується таким чином. Можливо також використання такого стилю для добування змінної із ФОРМИ, якщо ввімкнено *register_globals* у файлі конфігурації *php.ini*.

– *середній стиль: $_POST['varaible'],$_GET['variable'], $_REQUEST['variable']*. Використовується для отримання даних з полів ФОРМИ. В залежності від типу передачі даних, встановлюється POST або GET або REQUEST для обох випадків.

– довгий стиль: $HTTP_POST_VARS['variable'], *$HTTP_GET_VARS['variable']*.

– Найповніший стиль запису, найменш використовуваний. Починаючи з PHP 5.0.0 ці змінні можна вимкнути. Починаючи з PHP 6 ці змінні недоступні.

Приклад запису змінної з плавоючою точкою та рядкової змінної:
$*variable = 0.00; $variable1 = 'hello PHP-Master!';*
$variable2 = "hello PHP-Coder!";

З точки зору системи типізації, PHP є мовою програмування з динамічною типізацією. Немає необхідності явного визначення типу змінних, хоча така можливість існує. В разі звернення до змінної, інтерпретатор PHP трактує її тип відповідно до контексту. За необхідності можливе приведення змінної до певного типу за допомогою відповідних конструкцій мови. Це може знадобитись, якщо зважити, що значення змінної можуть трактуватись по-різному в залежності від її типу. Також можливе визначення типу відповідної змінної на певному етапі виконання сценарію. Імена змінних чутливі до регістру символів.

До базових типів належать булеві дані, цілі та дійсні числа із плаваючою комою, а також рядки. Булеві дані виражають істинність значення. Цілі числа можуть бути подані у вісімковому, десятковому та шістнадцятковому вигляді. Розмір цілого числа може змінюватись залежно від платформи, як правило, розрядність становить 32 біти. PHP не підтримує беззнакові цілі числа. Дійсні числа із плаваючою комою можуть бути подані в десятковій або експоненційній формі.

Робота з рядками

Рядки ділять на два класи — рядки, що підлягають аналізу, та ті, що не підлягають. Перший клас досліджується інтерпретатором на наявність посилань на інші змінні, і за умови їхньої наявності робиться підстановка значень у відповідне місце. Крім того, клас дозволяє проводити маніпуляції з керівними символами. Символ рядка може мати лише одне з 256 значень, але є можливість працювати з багатобайтовими символами. Доступ до символів рядка можливий з використанням синтаксису, схожого на доступ до елементів масивів.

PHP надає широкий спектр функцій для пошуку та заміни тексту в рядках. Для цього використовують як традиційний підхід, так і спеціальний, що базується на використанні регулярних виразів. При цьому в мові реалізована підтримка двох видів регулярних виразів — Perl-сумісні та POSIX-сумісні, що розрізняються за синтаксисом та особливостями роботи.

Змішані типи даних

До змішаних типів належать масиви, хеші та об'єкти. Масиви в сенсі мови є наборами змінних, що згруповані в єдину змінну. Вимога однотипності наповнення масивів не ставиться. Технічно, масиви — це впорядковані карти, що відображають ключові значення на позиції змінних даних. Вмістом значення, на яке вказує ключ може бути будь-чим, що можна подати у вигляді змінної. Не існує жодних обмежень, крім обсягу пам'яті, що накладаються на кількість ключів масиву.

Особливістю мови є відмова від рівномірного розподілу ключів масивів. Реалізовано і модель багатовимірних масивів, причому без явного обмеження глибини вкладеності. Корисною властивістю PHP є можливість асоціації масивів із функцією зворотного виклику. Ці функції дозволяють проводити дії над одним чи кількома масивами в пакетному режимі.

Також, існують два спеціальні типи даних — ресурс та NULL. Ресурс — спеціальна змінна, що містить посилання на зовнішній ресурс. Ресурси створюються та використовуються в спеціалізованих функціях. Оскільки тип містить спеціальні вказівники на відкриті файли, під'єднання та інше, то

будь-які дії над значенням ресурсу не мають сенсу.

Область видимості змінної — середовище, в якій вона визначена. Розрізняють локальні та глобальні змінні. За замовчуванням, всі змінні мають локальний характер дії. Виділяють особливий тип змінних — статичні, що дозволяє повторне звернення до змінної в певному сегменті коду, причому змінна буде зберігати попередньо отримане значення. Існує також поняття суперглобальних змінних, які є місцем збереження даних оточення або даних, отриманих ззовні. Підтримується концепція динамічних змінних та функцій.

Константи в PHP — ідентифікатори простих значень. Можливе визначення константи, причому після її оголошення стає неможливою зміна її значення чи анулювання. Константи можуть мати лише скалярні значення. Підтримується можливість отримання значення константи за динамічним ім'ям. Область видимості констант буде глобальною для сценарію та всіх під'єднаних компонентів. Також в ядрі мови визначено чимало системних констант.

Конструкції мови

Оператори в сенсі мови дозволяють виконувати відповідну дію над одним чи кількома операндами. Оператори бувають трьох типів — унарні, бінарні та тернарні. Оператори, як і в інших мовах характеризуються не лише дією, а й асоціативністю та пріоритетністю. Особливістю булевих операцій порівняння — розрізнення двох класів — з врахуванням типу і без врахування типу, при якому відбувається приведення до відповідного типу. Округлення відбуваються завжди в меншу сторону. В мові реалізовані особливі класи операторів — виконання, управління помилками та перевірки приналежності до класу.

Функції в сенсі мови є контейнерами коду, причому можливе включення інших функцій та класів. На цьому і базується можливість умовного визначення функції. В цьому випадку висувається вимога попередньої декларації викликаної функції, що не обов'язкове в інших випадках. Можливості перевизначення чи деактивації функції не існує.

Результат, який повертає функція може мати будь-який тип.

В мові реалізована функціональність посилань. Можливо створити скільки завгодно псевдонімів, що посилаються на єдиний сегмент даних. При вивільненні будь-якого з псевдонімів, сегмент даних залишається в пам'яті до моменту завершення сценарію або вивільнення усіх посилань.

Що стосується функцій в PHP, то замість прийнятого в багатьох мовах принципу перевантаження функцій, що дозволяє змінити хід виконання певної функції в залежності від типу та кількості переданих параметрів, використовується метод динамічних аргументів. Це дає змогу не визначати кількість параметрів для функцій при їх оголошенні, а працювати із тими аргументами, які були отримані на момент виклику функції. У тілі функції можливо отримати кількість переданих їй аргументів і проводити відповідні маніпуляції. При оголошенні функції звичайним чином, можливе задання значень аргументів за замовчуванням. Функції можуть повертати лише одне значення, проте це обмеження можна оминути, використавши не лише масиви, а й посилання. Передача аргументів за посиланням неможлива під час виконання та оголошення функції.

Після виконання сценаріїв, простір пам'яті, займаної ними очищується збирачем сміття. Проте, за необхідності можливе виконання очищення пам'яті від надлишкових сегментів даних під час виконання скриптів. Використання функцій очищення пам'яті є невиправданим, хоча така можливість існує.

Для побудови програмних комплексів можна використовувати модульний підхід, виконуючи розділення різнорідного коду. При потребі, можливе виконання під'єднання необхідних модулів, причому операція виконання може бути і умовною. Під'єднані до скрипта файли можуть повертати значення.

Додаткові можливості

Мова явно підтримує HTTP cookies відповідно до специфікацій Netscape. Це дозволяє проводити встановлення та читання невеликих сегментів даних на стороні клієнта.

PHP надає можливість організації роботи з користувачем протягом сеансів (сесій). В сесії можна зберігати різні дані, включаючи об'єкти.

Недоліки PHP

– Незручність дизайну мови
– Змінні з символом *«$»*
– Складні назви поширених функцій (html_entities_decode, mysql_select_db, nl2br тощо)
– Не підтримується Unicode в версіях до 6.0
– Непередбачуваність нових версій PHP.
– Відсутність єдиного драйвера для баз даних

1.7 Технологія MySQL

MySQL — це вільна система керування реляційними базами даних. У реляційній базі даних дані зберігаються не все скопом, а в окремих таблицях, завдяки чому досягається виграш в швидкості і гнучкості. Таблиці зв'язуються між собою за допомогою стосунків, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць.

Ця СКБД з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб- сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL — компактний багатонитковий сервер баз даних. Характеризується великою швидкістю, стійкістю і простотою використання.

MySQL володіє також рядом зручних можливостей, розроблених в тісному контакті з користувачами. Спочатку сервер MySQL розроблявся для управління великими базами даних з метою забезпечити вищу швидкість роботи в порівнянні з аналогами, що існували на той момент. І ось вже протягом декількох років даний сервер успішно використовується в умовах промислової експлуатації з високими вимогами. Не дивлячись на те, що MySQL

постійно удосконалюється, він вже сьогодні забезпечує широкий спектр корисних функцій. Завдяки своїй доступності, швидкості і безпеці MySQL дуже добре личить для доступу до баз даних по Internet.

Технічні можливості СУБД MYSQL MYSQL є системою клієнт-сервер, яка містить багатопотоковий sql-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також декілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API).

MySQL був розроблений компанією «TcX» для підвищення швидкодії обробки великих баз даних.

MySQL вважається гарним рішенням для малих і середніх застосувань. Вихідні коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX- системах, де є підтримка багатонитковості, що підвищує продуктивність системи в цілому.

Для некомерційного використання MySQL є безкоштовним.

Можливості сервера MySQL:

– простота у встановленні та використанні;

– підтримується необмежена кількість користувачів, що одночасно працюють із БД;

– кількість рядків у таблицях може досягати 50 млн.;

– висока швидкість виконання команд;

– наявність простої і ефективної системи безпеки.

Пристрій MYSQL

MYSQL складається з двох частин: серверною і клієнтською.

Сервер MYSQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MYSQL sql-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Структура MYSQL трирівнева: бази даних – таблиці – записи. Бази даних і таблиці MYSQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно – таблиця є сукупністю записів. А записи – це сукупність полів різного типа. Ім'я бази даних MYSQL унікально в межах системи, а таблиці – в межах бази даних, поля – в межах таблиці. Один сервер MYSQL може підтримувати відразу декілька баз даних, доступ до яких може розмежовуватися логіном і паролем. Знаючи ці логін і пароль, можна працювати з конкретною базою даних. Наприклад, можна створити або видалити в ній таблицю, додати записи і так далі. Зазвичай ім'я-ідентифікатор і пароль призначаються хостинг-провайдерами, які і забезпечують підтримку MYSQL для своїх користувачів.

Ліцензування

MySQL має подвійне ліцензування. MYSQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати вихідні тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії MySQL AB, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GPL.

Історія

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблицям, для яких використовувалися ISAM — підпрограми низького рівня. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL. Звідки походить назва «MySQL» — достеменно не відомо. Розробники дають два варіанти: або тому, що практично всі напрацювання компанії починалися з префікса My, або на честь дівчинки на ім'я My, дочки Майкла Монті Віденіуса, одного з розробників системи.

Логотип MySQL у вигляді дельфіна носить ім'я «Sakila». Він був

обраний з великого списку запропонованих користувачами «імен дельфіна». Ім'я «Sakila» було відправлено Open Source-розробником Ambrose Twebaze.

В січні-лютому 2008 Sun Microsystems придбала розробника системи керування базами данних MySQL за $1 млрд. Після поглинання у 2009 році Sun Microsystems компанією Oracle Corporation MySQL стала власністю Oracle.

Релізи:

- перший внутрішній випуск MySQL — 23 травня 1995
- версія для систем Windows ((Windows 95 і NT))— 8 січня 1998
- версія 3.23: бета від червня 2000, стабільний випуск у січні 2001
- версія 4.0: бета від серпня 2002, стабільний випуск у березні 2003 (unions, query cache)
- версія 4.1: бета від червня 2004, стабільний випуск у жовтні 2004 (r-trees, subqueries)
- версія 5.0: бета від березня 2005, стабільний випуск у жовтні 2005 (cursors, stored procedures, triggers, views, XA transactions)
- версія 5.1: розробка велась із листопада 2005, стабільний випуск в листопаді 2008
- версія 5.4: бета в квітні 2009
- версія 6.0: в розробці

Відгалуження MySQL

Продаж MySQL Sun Microsystems у 2008 і наступне поглинання Sun з боку Oracle було недоброзичливо сприйнято спільнотою вільного програмного забезпечення. Незалежними групами, в тому числі навколо колишніх розробників MySQL були створені відгалуження (форки) проекту, серед яких

- Drizzle
- MariaDB
- PersonaDB

Критика

MySQL версії 5.1 має 20 відомих серйозних помилок в додаток до 35 дефектів версії 5.0. Критичні помилки іноді не виправляються протягом тривалих періодів часу. Одним із прикладів є критична помилка, що відома з

2003 року.

MySQL показує низьку ефективність при використанні її як сховище даних, це частково через нездатність використовувати кілька процесорів для обробки одного запиту. До того ж, MySQL часто критикують за те, що ця СУБД має розходження зі стандартом SQL щодо трактування NULL значень і значення за замовчуванням.

Відомі користувачі

- Apple — для динамічних сторінок на Apple.com, для багатьох внутрішніх проектів
- Amazon.com — для багатьох внутрішніх проектів
- Cox Communications — четвертий за розміром провайдер кабельного телебачення в США, має понад 3,600 таблиць та 2 мільярди рядків даних у базах і виконує приблизно 2 мільйони вставок на годину.
- Digg.
- flickr
- Google — для програми AdWords пошукового рушія
- LiveJournal — приблизно 300 мільйонів переглядів сторінок на день
- MediaWiki і Вікіпедія
- NASA — конвертувала базовану на Oracle систему закупок на MySQL
- Nokia
- Yahoo!

MySQL 5.1

Версія MySQL 5.1 продовжує шлях до стандарту SQL:2003.

MySQL 5.1 містить такі нововведення:

- Сегментування — можливість розбити одну велику таблицю на декілька частин, розміщених в різних файлових системах, базуючись на визначеній користувачем функції. При деяких умовах це може дати серйозне збільшення продуктивності та, крім того, полегшує масштабування таблиць.
- Змінено поведінку ряду операторів для забезпечення більшої

сумісності зі стандартом SQL:2003.

- Порядкова реплікація (row-based реплікація), при якій в бінарний лог буде записуватись тільки інформація про реально змінені рядки таблиці замість оригінального (і, можливо, більш повільного) тексту запиту.
- Вбудований планувальник робіт, що періодично запускаються. По синтаксису додання задачі схоже на додання тригера до таблиці; по ідеології — на crontab.
- Додатковий набір функцій для обробки XML, реалізація підтримки XPath.
- Нові засоби діагностики проблем і утиліти для аналізу продуктивності.
- Утиліта mysql_upgrade, яка призначена для обновлення.
- API для плагінів.
- Новий рушій таблиць Maria (стійкий до збоїв клон MyISAM), який у 2010 був перейменований на Aria та став основою розділення MySQL від Монті Віденіуса під назвою MariaDB.

1.8 Веб-додаток phpMyAdmin

phpMyAdmin — веб-застосунок з відкритим кодом, написаний на мові PHP, представляє собою веб-інтерфейс для адміністрування СУБД MySQL. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць й баз даних. Додаток користується великою популярністю у веб-розробників, оскільки дозволяє керувати СУБД MySQL без безпосереднього вводу SQL команд, надаючи дружній інтерфейс.

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов’язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СУБД MySQL. Переважна більшість українських провайдерів використовують цей додаток як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Програма розповсюджується під ліцензією GNU General Public License і тому деякі інші розробники інтегрують його у свої розробки, наприклад XAMPP, Denwer.

Ініціатором є німецький розробник Тобіас Ратшіллер (Tobias Ratschiller) і базується, так само як і phpPgAdmin, на аналогічному додатоку MySQL-Webadmin, що виник у 1997 році.

Проект на даний момент часу локалізовано 57-ма мовами з підтримкою писемності як зліва направо, так і справа наліво.

Починаючи з версії 3.0.0, phpmyadmin приєднався до ініціативи Gophp5 і припинив підтримку написання коду під застарілі версії PHP і MYSQL; версія 3 і подальші вимагають наявність PHP 5.2 і MYSQL 5. При використанні із застарілими версіями PHP і MYSQL, виберіть попередню, але все ще підтримувану, вітку 2.x випусків.

phpMyAdmin виграв декілька призів. Серед інших, він був вибраний як краща програма на PHP, і щороку отримує приз співтовариства Sourceforge.net, як краща програма для системних адміністраторів.

При адмініструванні нашої бази даних ми користуватимемося версією *phpMyAdmin-4.7.3*

Логін і пароль для входу в admin частину:

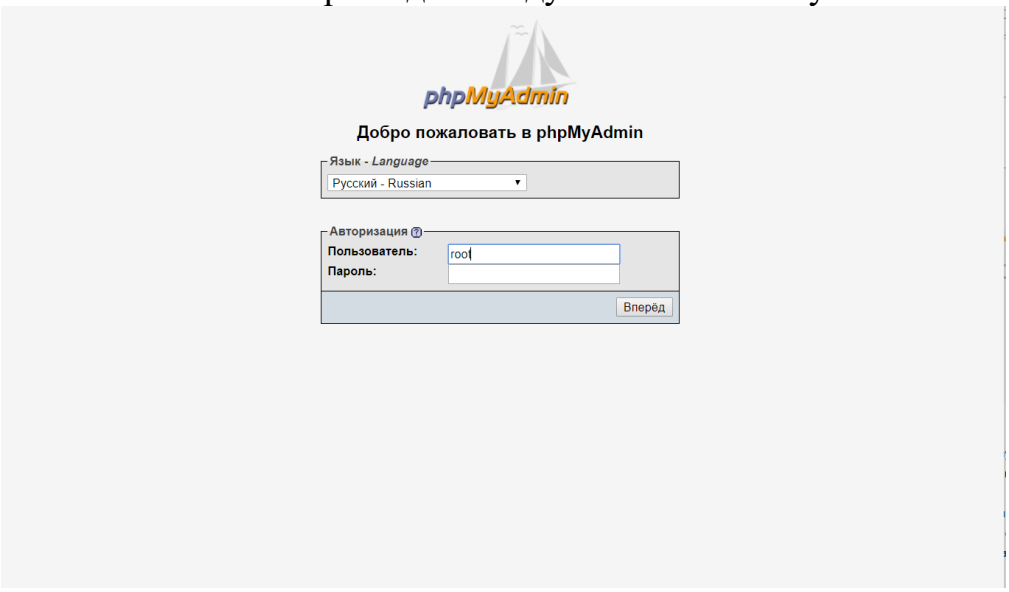


Рис.1.2. Вхід в admin частину

Адмінка:

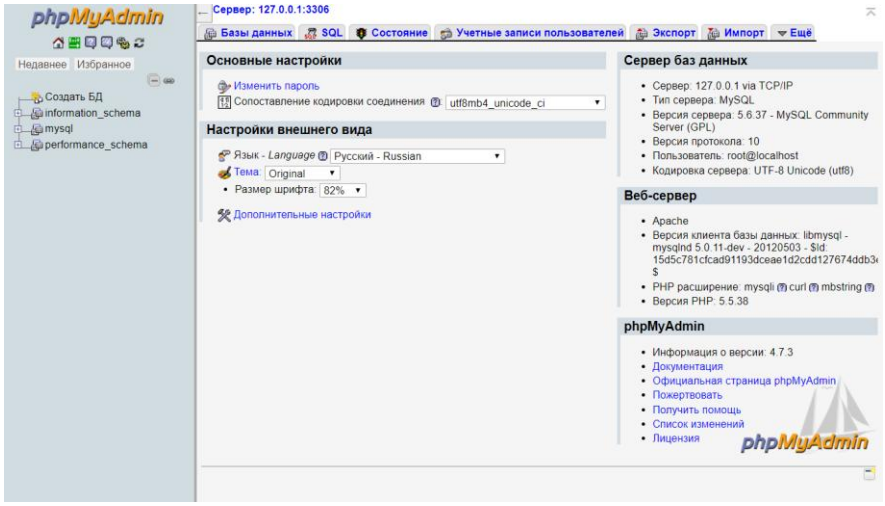


Рис.1.3. Admin частина

Можливості phpMyAdmin

1. Інтуїтивно зрозумілий інтерфейс.
2. Підтримка більшості функцій MYSQL.
3. Перегляд і видалення баз даних, таблиць, полів, індексів.
4. Створення, копіювання, видалення, перейменування і зміна баз даних.
5. Управління сервером, базами даних і таблицями.
6. Виконання, редагування і збереження будь-якого sql-запиту, включаючи

пакетні запити.

7. Управління користувачами MYSQL і їх привілеями.
8. Робота з процедурами, що зберігаються, і тригерами.
9. Підтримка імпорту даних з CSV і SQL.
10. Підтримка експорту в різні формати CSV, SQL, XML, PDF, Iso/iec 26300 -

Opendocument текст і таблиці, Word, Excel, LATEX та інші.

11. Адміністрування декількох серверів.
12. Генерування наочних схем баз даних у вигляді PDF.
13. Створення комплексних запитів.
14. Глобальний або частковий пошук в базі даних.

15. Трансформація даних в будь-який формат, використовуючи набір призначених функцій на зразок відображення blob-даних у вигляді малюнку або запиту для скачування.

Це не все, лише частина можливостей phpMyAdmin яких, втім, досить щоб пояснити його міжнародну популярність.

2. ПРОЕКТУВАННЯ СТРУКТУРИ ТА БАЗИ ДАНИХ

2.1 Організація діяльності Web-сайту

Даний web-сайт приставляє собою вигаданий інтернет – магазин комп'ютерної техніки який створений в навчальних цілях.

Отже функціонуючу структуру Інтернет-магазину представимо наступним чином (Рис.2.1):

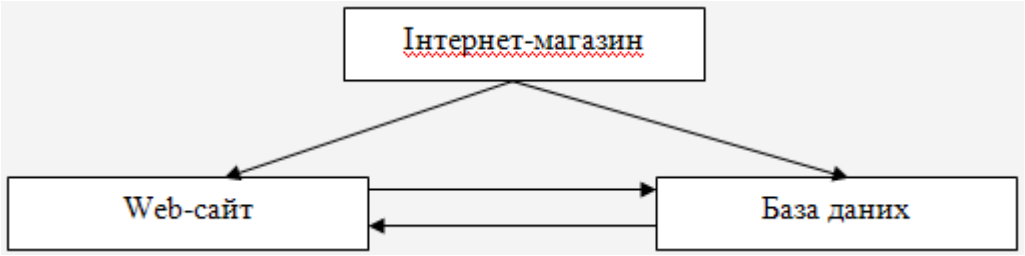
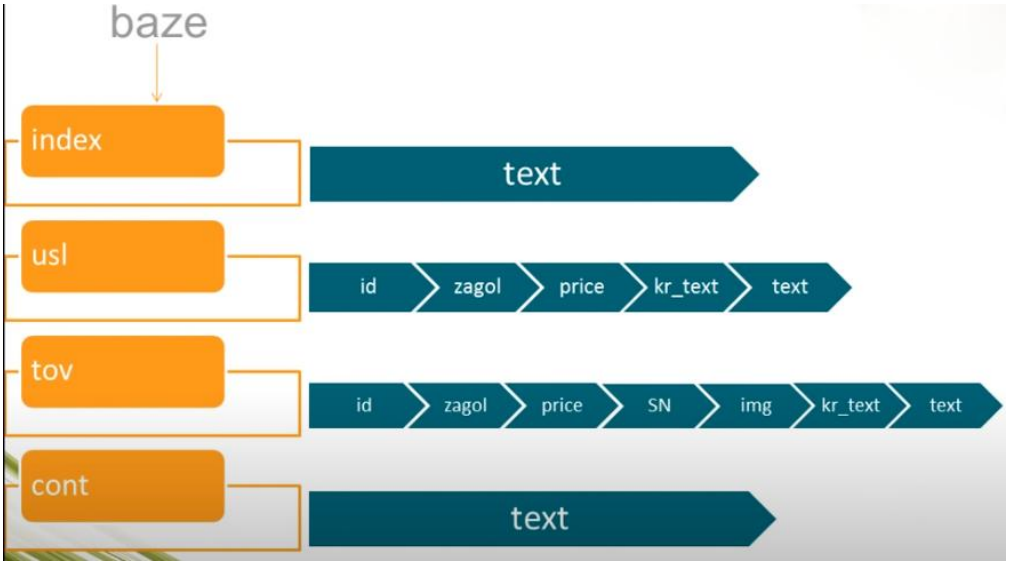


Рис.2.1. Функціонуюча структура Інтернет-магазину

Структуру бази даних представимо наступним чином (Рис.2.2):



Мал.2.2. Структура бази даних

База даних «Інтернет-магазину» складається з 4 таблиць (структуру бази даних можна переглянути на Рис.2.2):

– index (home);

– usl;

– tov;

– cont.

В таблиці «index (home)» лише одне поле «text» і в таблиці «cont» також одне поле «text». В таблиці «usl» п’ять полів таких як:

1. «id» - ідентифікатор послуги;
2. «zagol» - заголовок послуги;
3. «price» - ціна послуги ;
4. «kr_text» - короткий опис;
5. «text» - повний опис.

В таблиці «tov» сім полів:

1. «id» - ідентифікатор товару;
2. «zagol» - заголовок товару;
3. «price» - ціна товару;
4. «SN» - серійний номер;
5. «img» - посилання на зображення;
6. «kr_text» - короткий опис;
7. «text» - повний опис.

Структуру Web-сайту Інтернет-магазину представимо наступним чином:

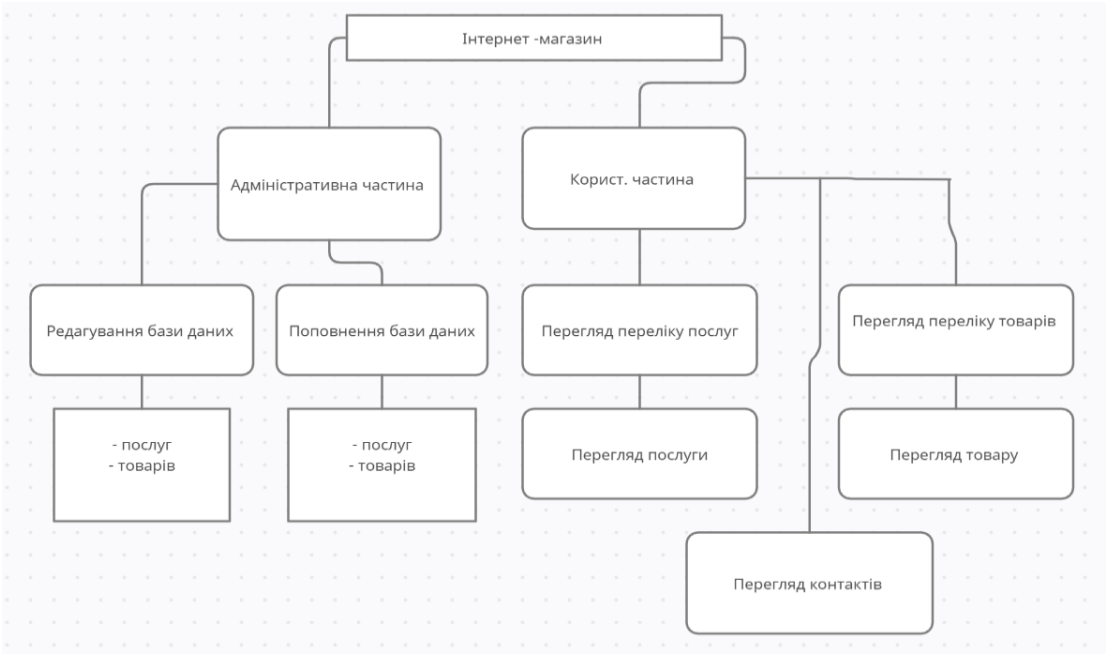


Рис.2.3. Структура Web-сайту Інтернет-магазину комп’ютерної техніки

Із структури видно (Рис.2.3), що пропонується 2 робочих частини: адміністративна та користувальницька.

Адміністративна частина дозволяє адміністратору здійснювати редагування бази даних та внесення додаткових елементів (послуг, товарів) в базу даних. Користувальницька частина даної можливості немає, тобто доступ до ресурсів закритий для загального огляду.

Користувальницька частина доступна всім користувачам. Дозволяє користувачам здійснювати перегляд інформації, перегляд послуг та товарів, які розміщені на web-сайті.

Для створення бази даних в phpMyAdmin, спочатку запустимо Open Server. Для цього двічі натискаємо на Open Server x64, в правому куті появляється іконка натискаємо на неї і вибираємо «Запустити» (Рис.2.4.)

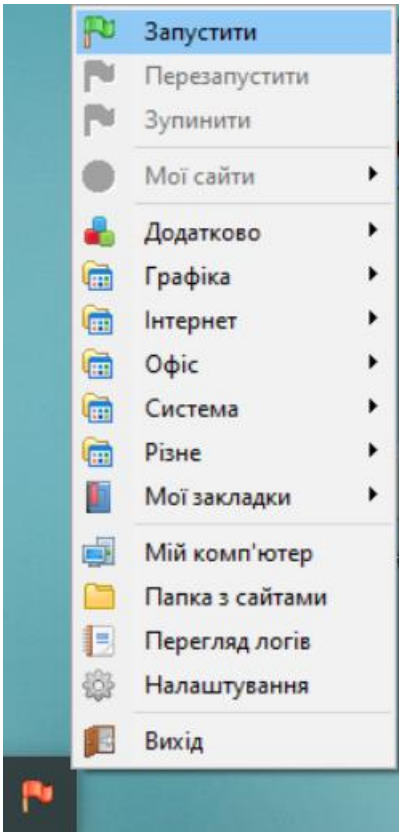


Рис.2.4. Запуск Open Server

Потім після запуску Open Server натискаємо пункт «Додатково» і обираємо phpMyAdmin (Рис.2.5.)

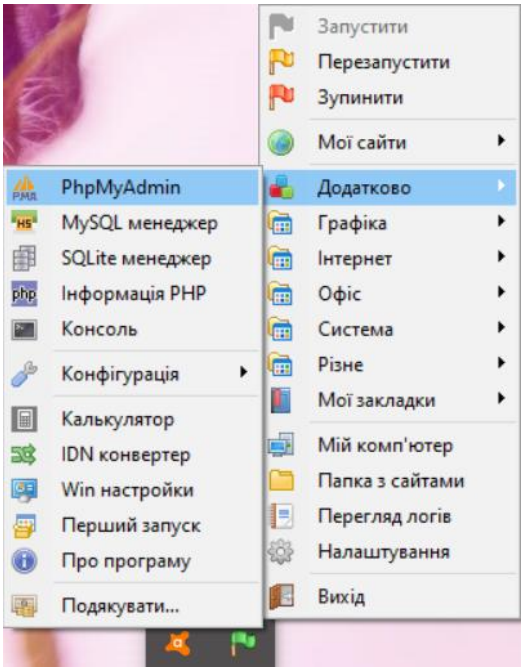
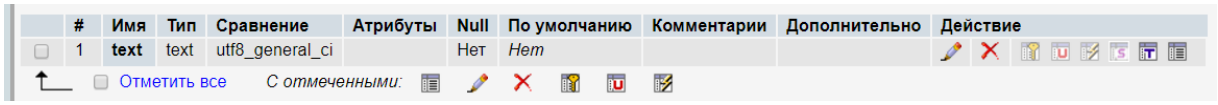


Рис.2.5. Створення бази даних

У відображеному вікні встановлюємо логін – «root» без пароля. Див. Розділ 1. (Рис.1.2.- Рис.1.3.) В admin частині створюємо базу даних під назвою «base», таблиці бази даних.

Структура таблиці «cont» (Рис.2.6.):

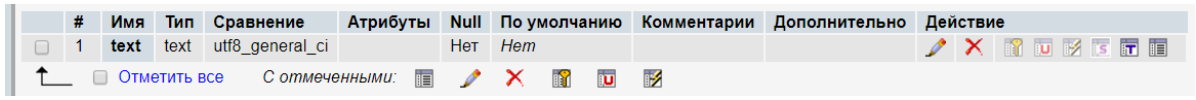


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	text	text	utf8_general_ci		Нет	Нем			

Отметить все С отмеченными:

Рис.2.7. Таблиця «cont»

Структура таблиці «home» (Рис.2.8.):



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	text	text	utf8_general_ci		Нет	Нем			

Отметить все С отмеченными:

Рис.2.8. Таблиця «home»

Структура таблиці «tov» (Рис.2.9.):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(6)			Нет	Нем		AUTO_INCREMENT	
2	zagol	varchar(100)	utf8_general_ci		Нет	Нем			
3	price	int(7)			Нет	Нем			
4	SN	varchar(30)	utf8_general_ci		Нет	Нем			
5	img	varchar(100)	utf8_general_ci		Нет	Нем			
6	kr_text	varchar(200)	utf8_general_ci		Нет	Нем			
7	text	text	utf8_general_ci		Нет	Нем			

Отметить все С отмеченными:

Рис.2.9. Таблиця «tov»

Структура таблиці «usl» (Рис.2.10.):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(6)			Нет	Нем		AUTO_INCREMENT	
2	zagol	varchar(100)	utf8_general_ci		Нет	Нем			
3	price	int(7)			Нет	Нем			
4	kr_text	varchar(200)	utf8_general_ci		Нет	Нем			
5	text	text	utf8_general_ci		Нет	Нем			

Отметить все С отмеченными:

Рис.2.10. Таблиця «usl»

У підсумку отримуємо такий склад бази даних "base" (Рис.2.11.)

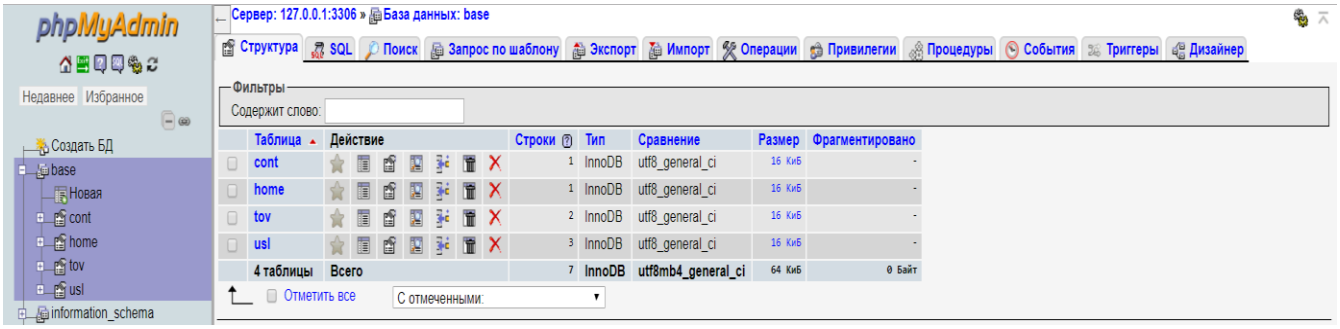


Рис.2.11. Структура бази даних "base" в phpMyAdmin

Надалі вся інформація по адмініструванню, зі зміни будь-якої інформації в інтернет-магазині буде фіксуватися в даній структурі бази даних.

3. РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ WEB-САЙТУ

Для відображення web-сайту запустимо Open Server. Див. Розділ 2. (Рис.2.5.) Потім натискаємо пункт «Мої сайти» і обираємо сайт під назвою «Pro» (Рис.3.1.)

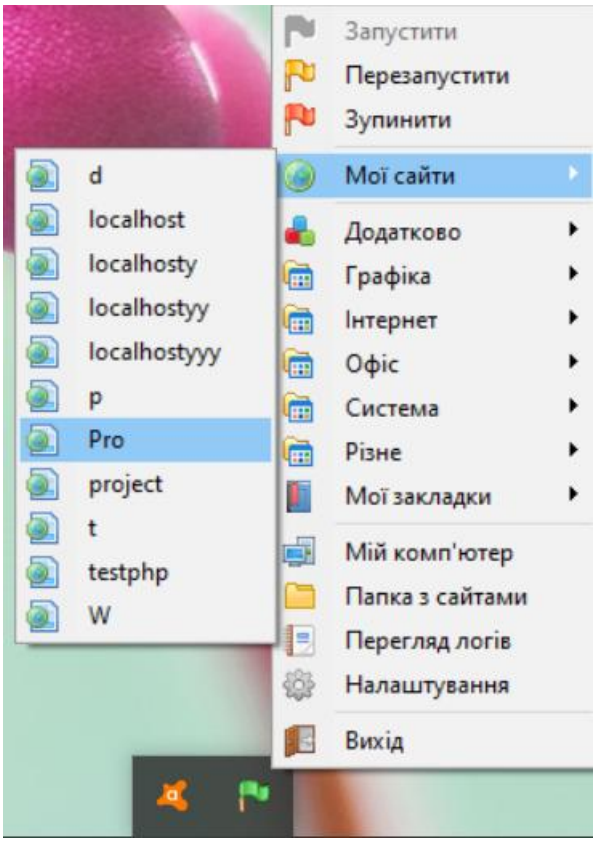


Рис.3.1. Запуск web-сайту

При завантаженні Web-сайту «комп’ютерної техніки» перед користувачем відкривається головна сторінка (Рис.3.2):

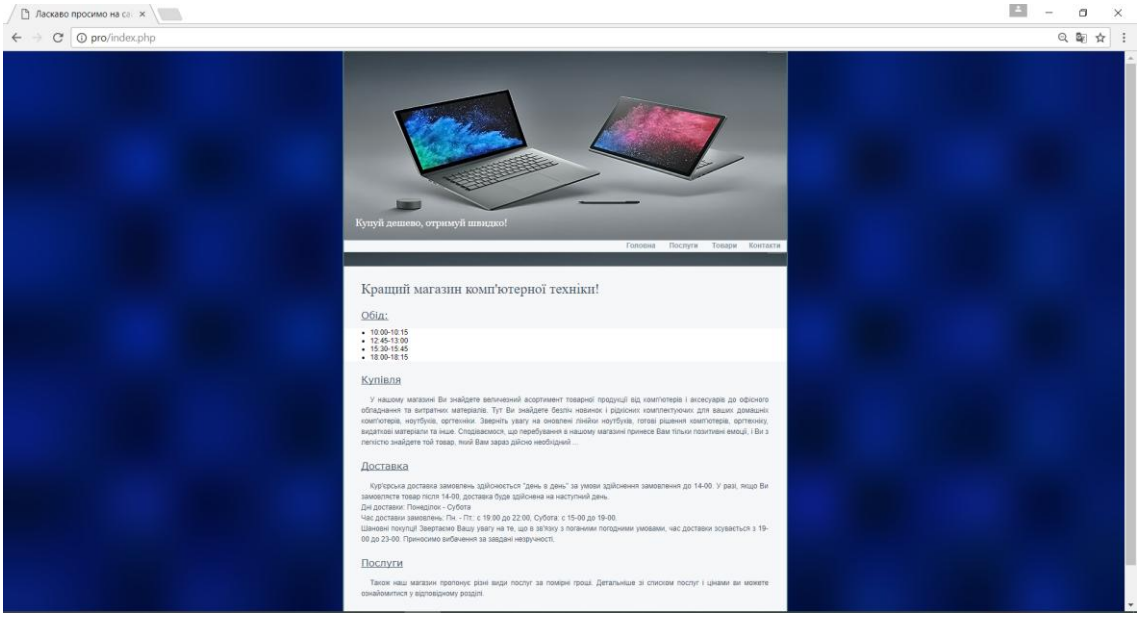


Рис.3.2. Головна сторінка Web-сайту «комп’ютерної техніки»

На даній сторінці подається актуальний перелік: обід, купівля, доставка, послуги. Розміщується інформація.

Інформаційне меню Web – сайту (Рис.3.3.):

Рис.3.3. Меню користувача

На сторінці «Послуги» представлені послуги які надає магазин. Інформація про послуги (Рис.3.4.):

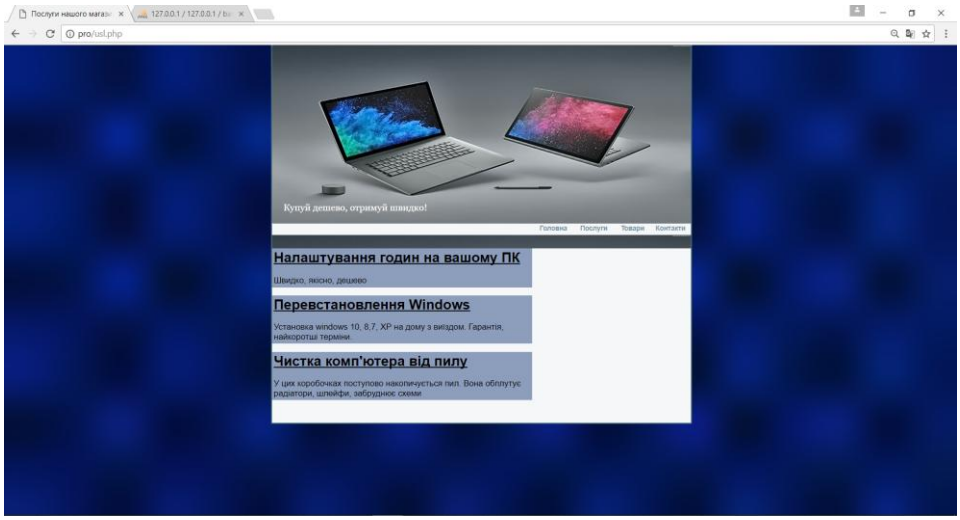


Рис.3.4. Послуги магазину

При натисканні на якусь представлену послугу відкривається інформація про дану послугу: «Назва», «Ціна», «Опис» (Рис.3.5.):

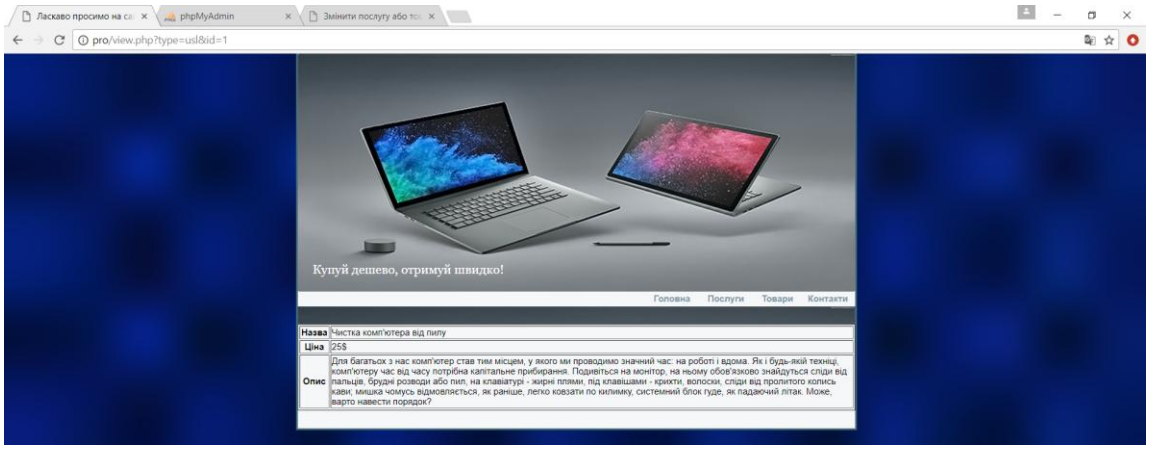


Рис.3.5. Інформація про послугу

На сторінці «Товари» представлені товари які пропонує інтернет – магазин. Інформація про товари (Рис.3.6.):

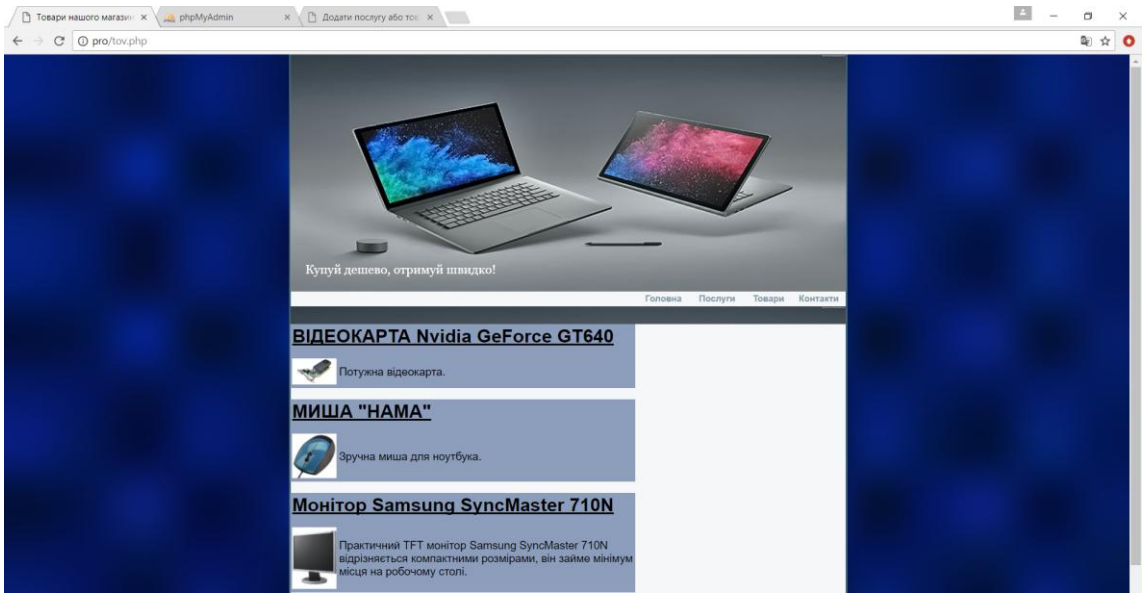


Рис.3.6. Товари магазину

Також при натисканні на якийсь представлений товар відкривається інформація про даний товар: «Назва», «Ціна», «Серійний номер», «Зображення», «Опис» (Рис.3.7.):

Рис.3.7. Інформація про товар

На сторінці «Контакти» представлена контактна інформація: адреса, телефони та Skype інтернет - магазину «компанії» (Рис.3.8.):

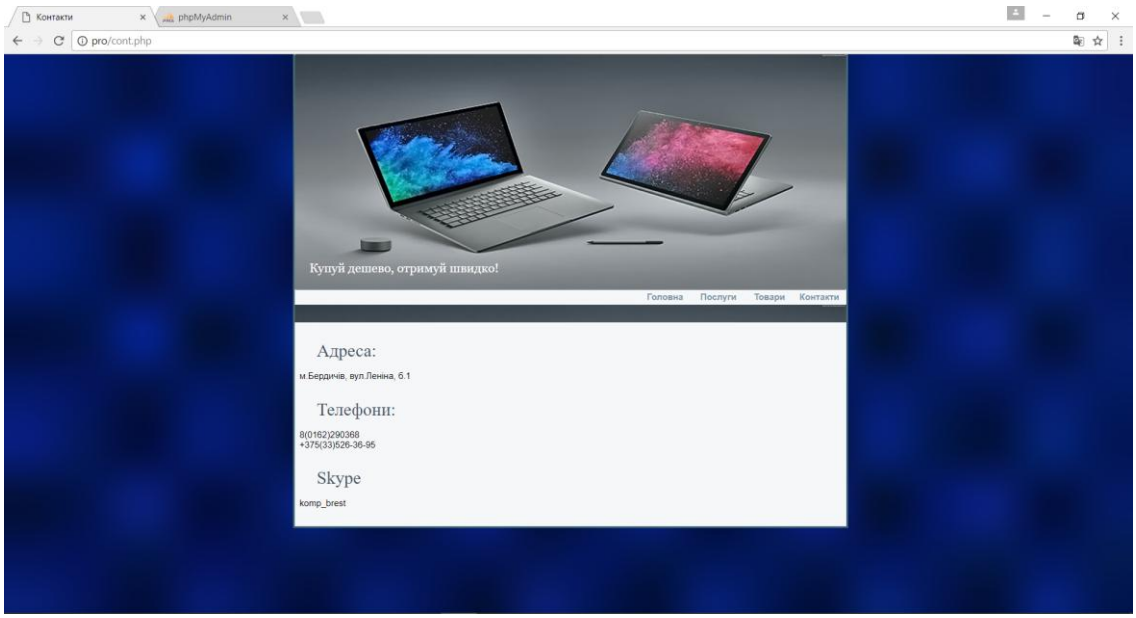


Рис.3.8. Контактна інформація інтернет – магазину

Перехід до адміністраторської частини сайту здійснюється при натисканні по адресному рядку і надрукувавши слово: «admin» (Рис.3.9.):

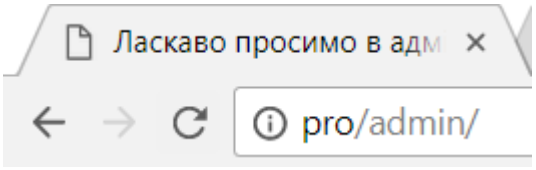


Рис.3.9. Перехід до адміністраторської частини сайту

При завантаженні «Адміністраторської частини» перед адміністратором відкривається головна сторінка (Рис.3.10):

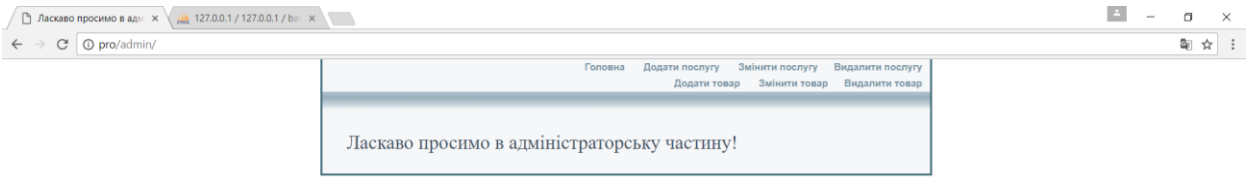


Рис.3.10. Головна сторінка «Адміністраторської частини» Web - сайту

На даній сторінці зображено меню «Адміністраторської частини» (Рис.3.11):

Рис.3.11. Меню «Адміністраторської частини»

На даній сторінці щоб додати послугу: вводимо «Назву послуги», «Ціну», «Короткий опис послуги», «Повний опис послуги» і натискаємо «Додати». Появляється повідомлення про те, що інформація вставлена успішно.

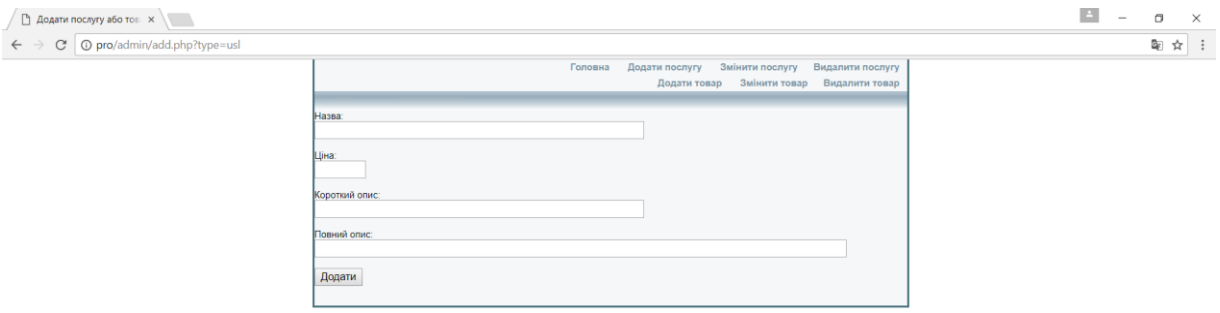


Рис.3.12. Додавання послуги

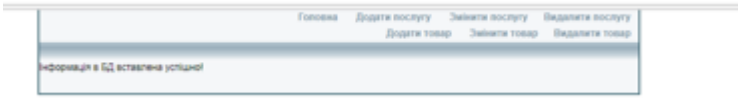


Рис.3.13. Інформація в БД вставлена успішно

Також щоб додати товар: вводимо «Назву товару», «Ціну», «Серійний номер товару», «Посилання на зображення», «Короткий опис товару», «Повний опис товару» і натискаємо «Додати». Появляється повідомлення про те, що інформація вставлена успішно. Якщо додавати інформацію некоректно появиться повідомлення: «При вставці інформації в БД сталася помилка!».

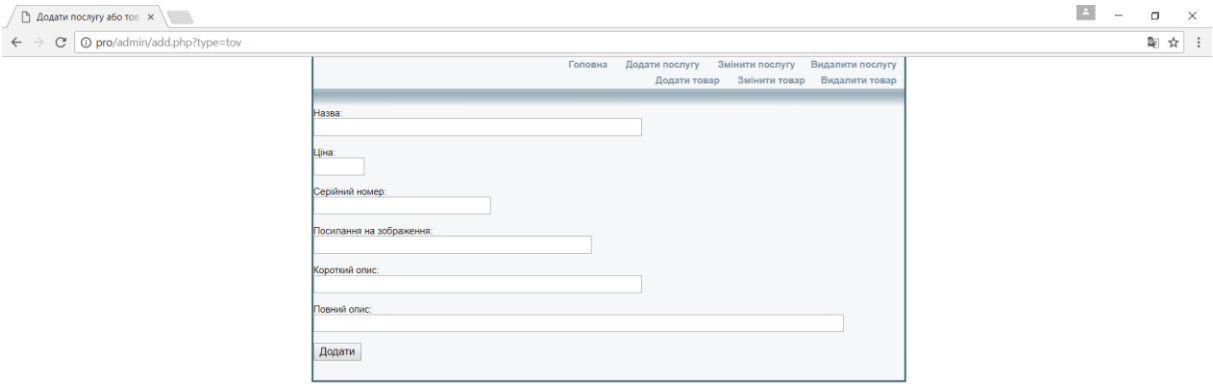


Рис.3.14. Додавання товару

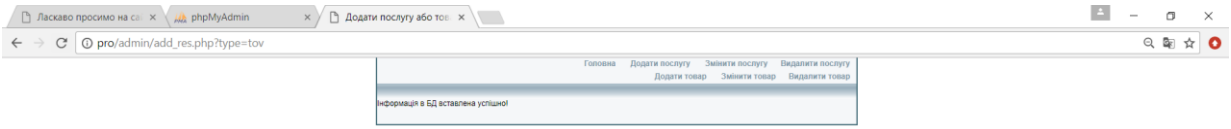


Рис.3.15. Інформація в БД вставлена успішно

На даній сторінці для того, щоб змінити (редагувати) послугу потрібно ввести ідентифікатор (номер) послуги, натиснути «Далі». (Рис.3.16.). Відкриється сторінка з інформацією даної послуги (Рис.3.17.), яку можемо змінити (редагувати). Після того як ми змінили інформацію про дану послугу натискаємо «Змінити». Далі появляється повідомлення про те, що інформацію в БД змінено успішно. (Рис.3.18.). Якщо змінювати інформацію про послугу не коректно появится повідомлення: «При змінні інформації в БД сталася помилка!».

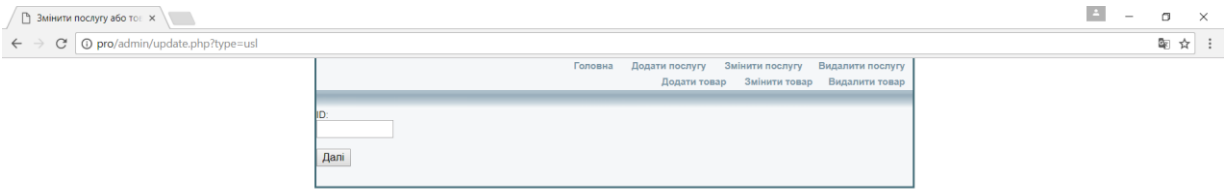


Рис.3.16. Введення «ID» послуги

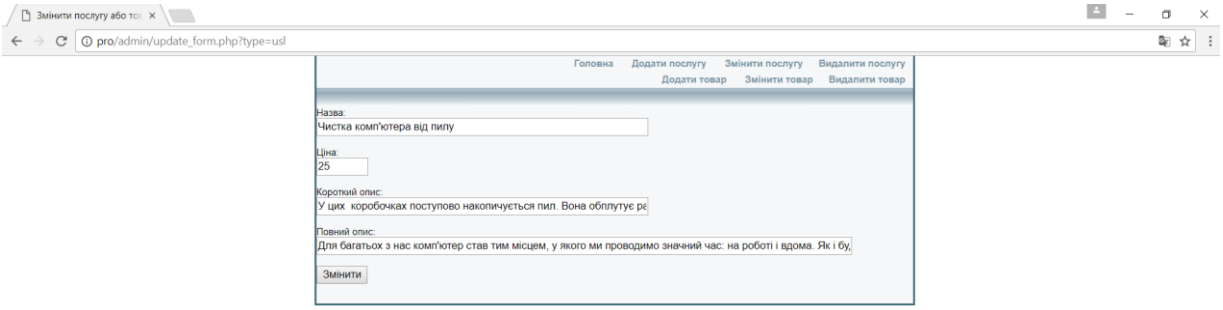


Рис.3.17. Зміна (редагування) інформації про послугу

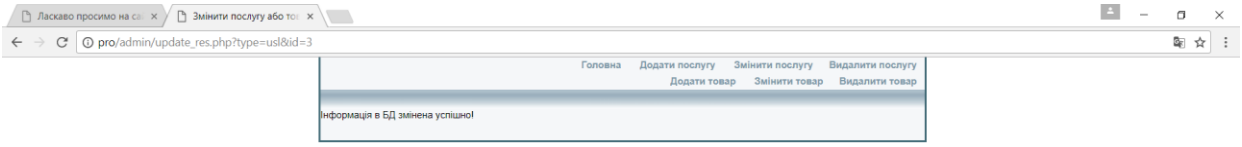


Рис.3.18. Інформацію в БД змінено успішно

Для того, щоб змінити (редагувати) товар: також вводимо ідентифікатор (номер) товару, натискаємо «Далі». (Рис.3.19.). Відкривається інформація про даний товар (Рис.3.20.), змінюємо інформацію про товар і натискаємо «Змінити». Появляється повідомлення про те, що інформацію в БД змінено успішно (Рис.3.21.). Якщо зміннювати інформацію про товар некоректно появиться повідомлення: «При змінні інформації в БД сталася помилка!».

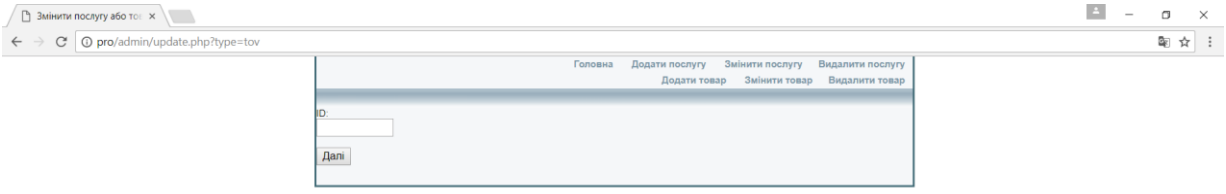


Рис.3.19. Введення «ID» товару

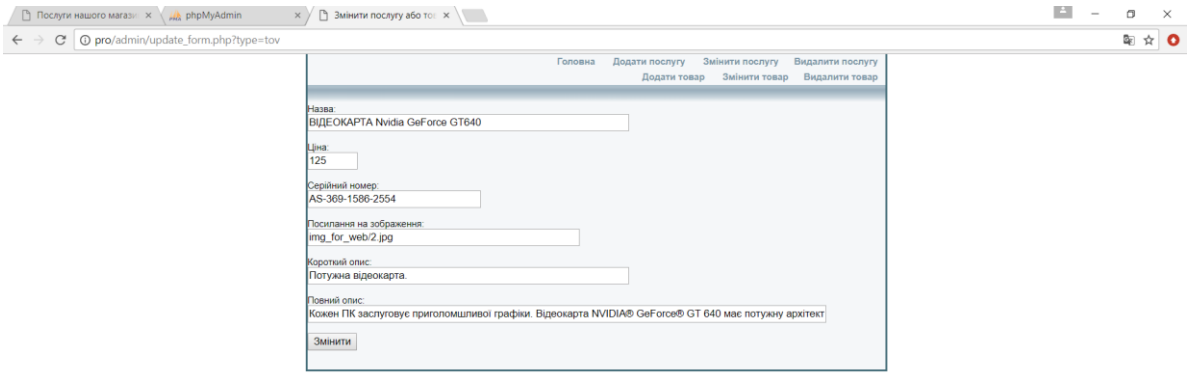


Рис.3.20. Зміна (редагування) інформації про товар

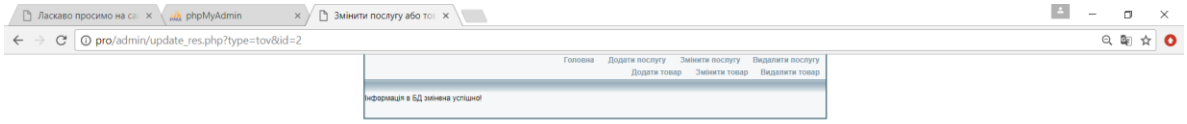


Рис.3.21. Інформацію в БД змінено успішно

При видаленні послуги вводимо: ідентифікатор (номер) послуги і натискаємо «Видалити» (Рис.3.22.). Також дану операцію проводимо і при видаленні товару (Рис.3.23.). Якщо видаляти інформацію некоректно появиться повідомлення: «При видаленні інформації в БД сталася помилка!».

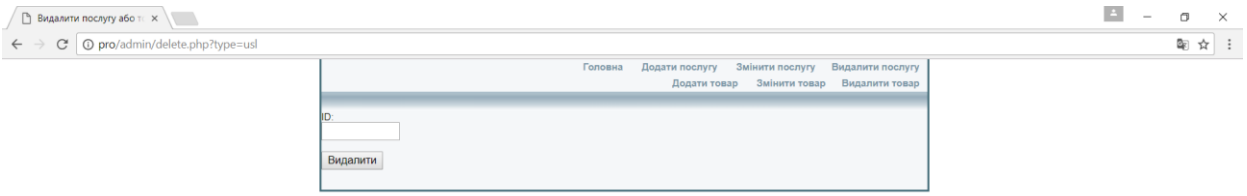


Рис.3.22. Видалення послуги

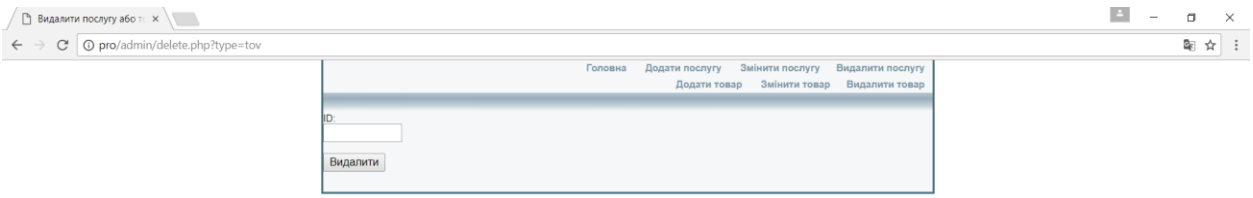


Рис.3.23. Видалення товару

4. ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

У наші часи науково-технічний прогрес вніс серйозні зміни до умов виробничої діяльності працівників розумової праці. Їх праця стала інтенсивнішою, напруженішою, що вимагає значних витрат розумової, емоційної і фізичної енергії. Це зумовило потребу у комплексному вирішенні проблем ергономіки, гігієни і організації праці, регламентації режимів праці і відпочинку.

Велике значення має раціональна конструкція і розташування елементів робочого місця, яке є важливим для підтримання оптимальної робочої пози оператора.

4.1 Виявлення шкідливих та небезпечних факторів

Відповідно до дипломного проекту експлуатація програмного комплексу та додаткової офісної техніки передбачається в приміщеннях типу обчислювальних центрів і з виробництвом не пов’язана.

Характерними особливостями обчислювальних центрів є відносно невелика площа робочих приміщень, системи кондиціювання повітря для відведення тепла від комп’ютерів, наявність розгалуженої електромережі для живлення комп’ютерів і периферійної техніки. Такі особливості приміщень можуть спричинити потрапляння користувачів системи під дію несприятливих факторів. Ці фактори можуть бути небезпечними та шкідливими. Вплив на людину шкідливого фактора може призвести до професійного захворювання. Випадок впливу на працівника небезпечного фактора при виконанні ним завдання керівника чи трудових зобов’язань називається нещасним випадком.

Вкажемо фактори, які характерні для нашого типу приміщення.

Шкідливі фактори і небезпечні фактори і їх припустимі рівні з посиланням на нормативні документи наведені у таблиці 4.1.

Таблиця 4.1 Шкідливі фактори і небезпечні фактори

№ п/п	Шкідливі фактори	Джерело шкідливого фактору	Характеристика	Нормативні документи
1	2	3	4	5
1	Шум (для програмістів комп'ютерів)	Системний блок, принтер, монітор, клавіатура	Рівень сили звуку L = 50 дБа	ДСанПіН 3.3.2.007-98, ГОСТ 12.1.003-83
2	Електромагнітне поле	Комп'ютер	Допустимі параметри поля: за електричною складовою (Е-10В/м), за магнітною складовою (Н=0,3 А/м)	ДСанПіН 3.3.2.007-98 [14], НПАОП 0.00-1.28-10
3	Несприятливий мікроклімат приміщення (категорія легка1)	Тепловипро-мінюючі прилади: монітор, системний блок	Оптимальні умови:t =22-24ºС в холодний і перехідний період, відносна вологість W= =40-60 %, швидкість руху повітря V = 0,1 м/с; в теплий період: t = 23-25ºС, відносна вологість W = 40-60 %, швидкість руху повітря V= 0,1 м/с	ГОСТ 12.1.005-88, ССБТ [11], ДСанПіН 3.3.2-007-98
4	Недостатнє освітлення (клас робіт дуже високої	Освітлювальні прилади	1)штучне освітлення Е=500лк. 2)природнє	ДБН В.2-5-28-2006

№ п/п	Шкідливі фактори	Джерело шкідливого фактору	Характеристика	Нормативні документи
	точності)		освітлення $К_{по}$=1,5 %	
5	Ураження електричним струмом	Ланцюги електроживлення, освітлення, порушення електроізоляції обладнання (системний блок, монітор)	f = 50 Гц U = 220 В I = 0,1 А P = 200-250 Вт	ГОСТ 12.1.038-82 ССБТ ПУЕ
6	Виникнення пожежі	Несправність електромережі (ЕОМ, зовнішні пристрої)	Підвищення температури, ступінь вогнестійкості II, категорія В	ГОСТ 12.1.004-85 ССБТ

4.2 Природне освітлення. Розрахунок природного освітлення

Дано: Офіс комп'ютерної компанії находиться в Києві. Геометричні розміри приміщення: довжина приміщення L = 12 м; глибина В = 6 м; висота Н = 3,3 м. Висота від робочої поверхні до верху вікна h1 = 1,2 м.

Коефіцієнти відбиття стелі, стін і підлоги: 80%, 40%, 10%.

Завдання: Розрахувати потрібну площу світлових прорізів.

Розрахуємо площу світлових прорізів при бічному висвітленні приміщення по формулі :

$S_o = (S_п * e_н * k_з * k_{буд} * η)/ (100 * t_o * r)$, де:

S_o– розрахункова площа світлових прорізів;

$S_п$– площа підлоги приміщення (72$м^2$);

$e_н$– нормоване розрахункове значення КПО;

$k_з$ – коефіцієнт запасу = 1,5;

$k_{буд}$– коефіцієнт, який враховує затінення сусідніми будівлями, приймається в межах 1…1,5;

η – світлова характеристика вікон;

r – коефіцієнт, що враховує підвищення КПО при бічному висвітленні завдяки світлу, відбитому від поверхонь приміщення і підстильного шару, що прилягає до будинку.

t_o – загальний коефіцієнт світлового пропускання, що визначається по формулі:

$t_o = t_1 * t_2 * t_3 * t_4 * t_5$, де:

t_1 – коефіцієнт світлового пропускання матеріалу склопакету, скло подвійне (0,8);

t_2 – коефіцієнт, що враховує втрату світла в плетіннях світлового пролому, дерев'яна одинарна рама (0,75);

t_3 – коефіцієнт, що враховує утрати світла в несущих конструкціях (1);

t_4 – коефіцієнт, що враховує утрати світла в сонцезахисних пристроях, штори (1);

t_5 – коефіцієнт, що враховує утрати світла в захисній сітці, установлюваної під ліхтарями (0,9).

t_o=0,8*0,75*1*1*0,9=0,54

Визначаємо значення КПО для м. Києва за формулою: $e_N = e_n * m_N$, де:

e_n – нормоване значення КПО згідно ДБН В.2.5-28-2006 «Природне та штучне освітлення» при боковому освітленні при роботах середньої точності; m_N– коефіцієнт світлового клімату. Визначаємо нормоване значення коефіцієнта природної освітленості для району розташування. Для Києва при орієнтації вікон на південь:

$e_N = 0,85 * 1,5 = 1,275\%$

Знаходимо світлову характеристику світлових отворів:

при L / B = 12: 6 = 2; B / h1 = 6: 1,2 = 5; η = 11.

Визначаємо коефіцієнт *r* - для бічного освітлення по таблиці.

Для цього знаходимо:

- відношення глибини приміщення до висоти від рівня умовної робочої поверхні до верху вікна - B / h1 = 5;
- відношення довжини приміщення до його глибини - L / B = 2;
- величину середньозваженого коефіцієнта відображення $\rho_{сер}$ стелі, стін і підлоги, при площі стелі S1 = 12 * 6 = 72 м2;
- площі стін S2 = 2 * 6 * 3,2 + 2 * 10 * 3,2 = 102.4 м2;
- площі підлоги S3 = 12 * 6 = 72 м2

$$\rho_{ср}=\frac{\rho_1 \cdot S_1+\rho_2 \cdot S_2+\rho_3 \cdot S_3}{S_1+S_2+S_3},$$

$$\rho_{сер} = \frac{72 * 80 + 40 * 102.4 + 10 * 72}{72 + 102.4 + 72} = \frac{10576}{246.4} = 43\%$$

Згідно з цим, по таблиці знаходимо r = 4,1.

Визначаємо площу світлових прорізів за формулою:

$$S_o = (S_n * e_н * k_з * k_{буд} * \eta)/ (100 * t_o * r) = \frac{72 * 1,275 * 1,5 * 1,2 * 11}{100 * 4,1 * 0,54} = 8,2 \text{ м}^2$$

Отже, для дотримання норм роботи в офісі комп'ютерної фірми, в приміщенні офісу потрібно встановити 2 вікна з розмірами 2,4м*1,8м.

Сумарна площа вікон (світлових прорізів):

$S_{вік.}$ = (2,4*1,8)*2 = 4.32*2=8,64 м^2

Площа запропонованих вікон задовольняє потребу у кількості світла і співпадає з визначеною вище площею світлових прорізів.

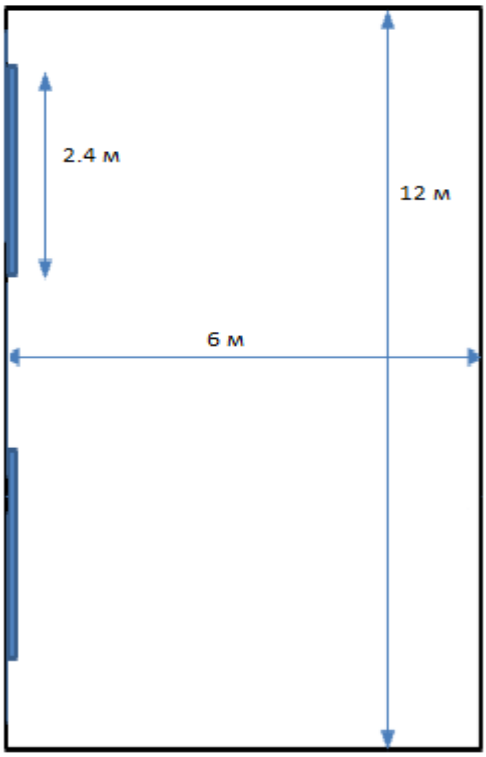


Рис. 4.1 Схематичне зображення вікон в кімнаті на плані приміщення

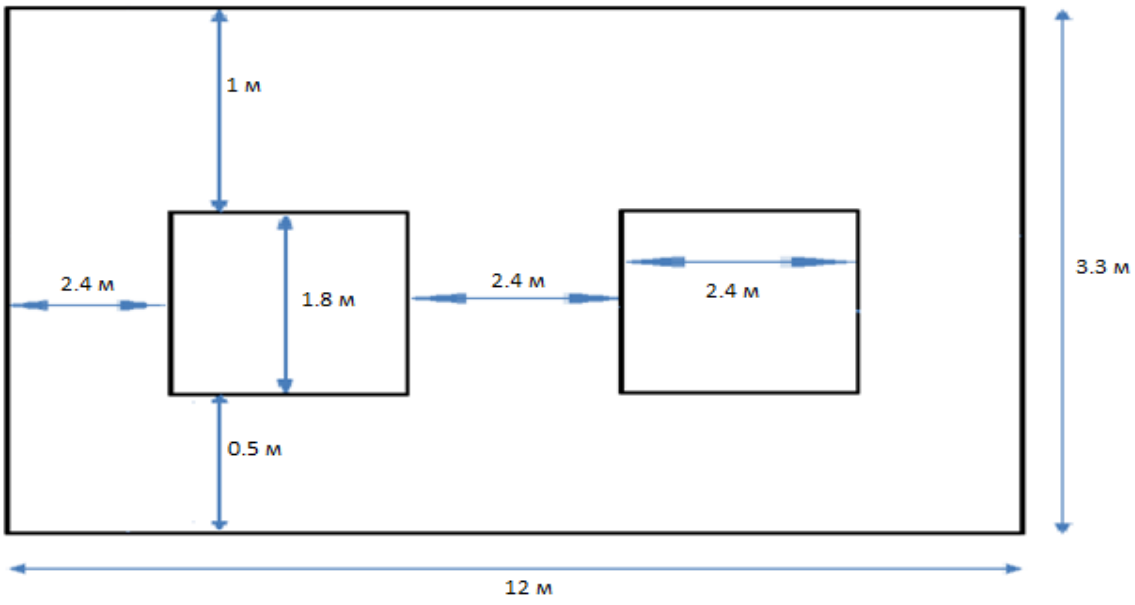


Рис. 4.2 Схематичне зображення вікон в кімнаті в поперечному розрізі

Визначимо геометричний коефіцієнт природної освітленості визначається за допомогою графіків Данилюка для розрахунку n1 та n2, де:

n1 – кількість променів за графіком I, які проходять від неба через світлові прорізи в розрахункову точку на поперечному розрізі приміщення;

n2 – кількість променів за графіком II, які проходять від неба через світлові прорізи в розрахункову точку на плані приміщення.

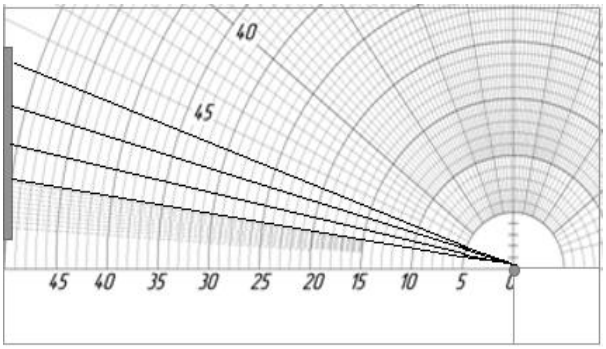


Рис. 4.3 Кількість променів, які проходять від неба через світлові прорізи в розрахункову точку на поперечному розрізі приміщення

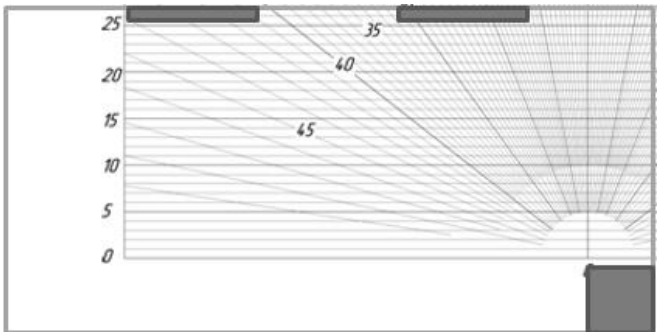


Рис. 4.4 Кількість променів, які проходять від неба через світлові прорізи в розрахункову точку на плані приміщення

Як можна побачити, на рис.8.3 та 8.4, n1 = 4 і n2 = 20

Розрахуємо геометричний КПО:

$$\varepsilon_0 = \big((n1 * n2) * 0{,}01\big) * 2 = \big((4 * 20) * 0{,}01\big) * 2 = 1{,}6$$

КПО при боковому освітленні:

$$e_р = \frac{\varepsilon_0 * q_3 * r * \tau_0}{K_3} = \frac{1{,}6 * 0{,}64 * 4{,}1 * 0{,}54}{1{,}5} = 1{,}51$$

$1{,}51\% > e_N = 1{,}35\%$

Отже, запропоновані вище параметри вікон повністю задовольняють вимоги офісу для програмістів.

Прямого світла відбитого від протилежного будинку немає, через те, що будівля стоїть відокремлено.

4.3 Штучне освітлення. Розрахунок штучного освітлення

Освітлення робочого місця нормується згідно з Державними будівельними нормами України: ДБН В.2.5-28-2006 «Інженерне обладнання будинків і споруд. Природне і штучне освітлення».

Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 500 лк.

Перевіримо освітленість робочого місця користувача ПК на відповідність розряду зорової роботи. Для штучного освітлення у приміщенні використовуються люмінесцентні лампи.

Розрахунок штучного освітлення проведемо для кімнати площею 72 м2, ширина якої складає 12м, довжина – 6м, висота – 3,3м.

Скористаємося методом використання світлового потоку. Для визначення потрібної кількості світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = \frac{E \cdot K \cdot S \cdot Z}{\eta}, \text{ де}$$

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; E = 500 Лк;

S – площа освітлюваного приміщення (у нашому випадку S=72 м2);

Z – відношення середньої освітленості до мінімальної (Z = 1,1 для люмінесцентних ламп);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку K = 1,5);

η – коефіцієнт використання світлового потоку, (виражається відношенням світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп, і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін (ρст.) і стелі (ρстелі)), значення коефіцієнтів дорівнюють ρст = 40% і ρстелі=60%.

Розрахункова висота підвісу h світильників задається, як правило, розмірами приміщення.

Розрахуємо розрахункову висоту підвісу, м:

$h = H - h_{СВ} - h_{РП}$,де:

Н – висота приміщення, м; hсв – висота підвісу світильника (від перекриття, "звис" світильника, приймається в межах від 0, при установці світильників на стелі, до 1,5 м), м ; hрп – висота робочої поверхні над підлогою, м (hрп = 1.2 м).

Підставивши значення отримаємо:

h=3,3 – 1,2 - 0,2=1,8 (м)

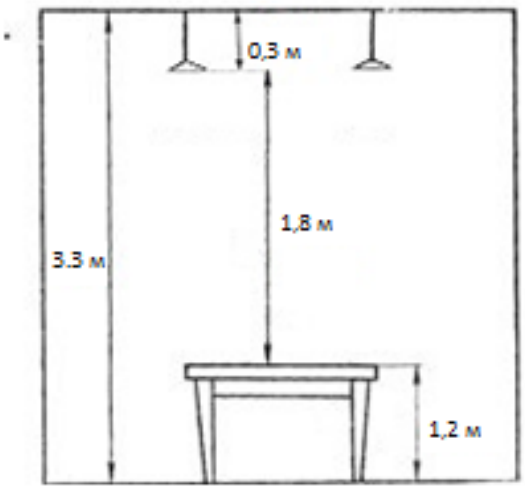


Рис. 4.5 Поперечний розріз приміщення

Обчислимо індекс приміщення за формулою:

$I = \frac{S}{h(A+B)}$, де

S – площа приміщення, S = 72 м2;

h – розрахункова висота підвісу, h = 1,8 м; А – ширина приміщення, А =

12 м; В – довжина приміщення, В = 6 м.

Підставивши значення отримаємо:

$$I = \frac{72}{1{,}8 * (6 + 12)} = 2{,}22$$

Знаючи індекс приміщення I, знаходимо η =0,34.

Підставимо всі значення у формулу для визначення світлового потоку F :

$$F = \frac{500 * 1{,}1 * 72 * 1{,}5}{0{,}34} = 174706 Лм$$

Для освітлення використані люмінесцентні лампи типу ЛБ 40-1, світловий потік яких F = 4320 Лм. Розрахуємо необхідну кількість ламп у світильниках за формулою:

$N = \frac{F}{Fл}$, де N – кількість ламп, що визначається; F - світловий потік, F = 174706; Fл- світловий потік лампи, Fл = 4320 Лм

$$N = \frac{174706}{4320} = 40.1 \approx 40$$

В приміщенні використовуються світильники типу ОД. Кожен світильник комплектується двома лампами. Тобто, необхідно використовувати 20 світильників із 40 працюючими лампами в них. Схема розташування світильників зображена на Рис. 4.6.

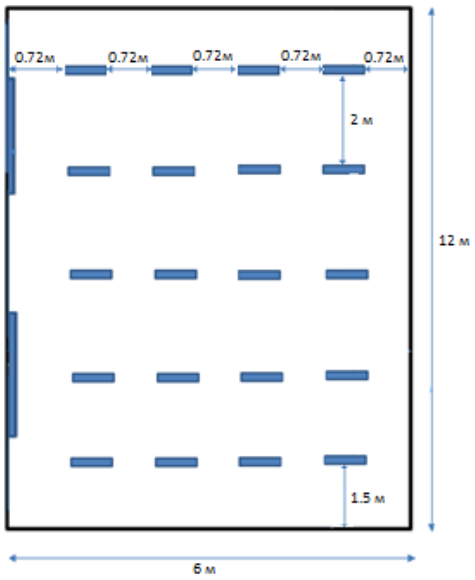


Рис. 4.6 Схема розташування світильників

4.4 Шум. Розрахунок шуму

Шум погіршує умови праці надаючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, зниження апетиту, біль у вухах і т.п. Такі порушення в роботі ряду органів і систем організму людини можуть викликати негативні зміни в емоційному стані людини аж до стресових. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється втома у зв'язку з підвищеними енергетичними витратами і нервово-психічним напруженням, погіршується мовна комутація. Все це знижує працездатність людини і її продуктивність, якість і безпеку праці. Тривала дія інтенсивного шуму вище 80 дБ на слух людини приводить до його часткової або повної втрати.

Рівень шуму на робочому місці математиків-програмістів і операторів відеоматеріалів не повинен перевищувати 50дБА. Для зниження рівня шуму стіни і стеля приміщень, де встановлені комп'ютери, можуть бути облицьовані звукопоглинальними матеріалами. Рівень вібрації в приміщеннях обчислювальних центрів може бути понижений шляхом встановлення устаткування на спеціальні віброізолятори.

Для вирішення питань про необхідність і доцільність зниження шуму необхідно знати рівні шуму на робочому місці оператора.

Рівень шуму, що виникає від декількох взаємопов’язаних джерел, що працюють одночасно, підраховується на підставі принципу енергетичного підсумовування рівня інтенсивності окремих джерел:

$$L_{\text{сер}} = 10lg \sum_{i=1}^{n} 10^{0,1 \times L_i},$$

де Li - рівень звукового тиску і-го джерела шуму;

n - кількість джерел шуму.

Отримані результати розрахунку порівнюється з допустимим значенням рівня шуму для даного робочого місця. Якщо результати розрахунку вище

допустимого значення рівня шуму, то необхідні спеціальні заходи щодо зниження шуму. До них відносяться: облицювання стін і стелі залу звукопоглинальними матеріалами, зниження шуму в джерелі, правильне планування обладнання і раціональна організація робочого місця оператора.

Рівні звукового тиску джерел шуму, що діють на оператора на його робочому місці. Рівні звукового тиску різних джерел.

Таблиця 4.2 Рівні звукового тиску різних джерел.

Джерело шуму	Рівень шуму, дБ
Жорсткий диск	20
Кулери	35
Монітор	15
Клавіатура	20
Принтер	45

Зазвичай робоче місце оператора оснащене наступним обладнанням: вінчестер в системному блоці, кулери систем охолодження ПК, монітор, клавіатура і принтер. Підставивши значення рівня звукового тиску для кожного виду обладнання у формулу, отримаємо:

$$L_{\Sigma} = 10 \cdot \lg (10^{2} + 10^{3,5} + 10^{1,5} + 10^{2} + 10^{4,5}) = 45{,}44 \text{ дБ}$$

Отримане значення не перевищує допустимий рівень шуму для робочого місця оператора, рівний 50 дБ (ГОСТ 12.1.003-83). І якщо врахувати, що навряд чи такі периферійні пристрої як сканер і принтер будуть використовуватися одночасно, то ця цифра буде ще нижчою. Крім того при роботі принтера безпосередню присутність оператора необов'язково, тому що принтер обладнаний механізмом авто подачі аркушів.

ВИСНОВКИ

Метою дипломного проекту є – показати застосування технологій PHP і MySQL при створенні Web – сайту комп'ютерної техніки.

Web-сайт являє собою вигаданий інтернет – магазин комп'ютерної техніки, який створений в навчальних цілях. Даний сайт розбитий на блоки, тобто кожен елемент сайту лежить в своєму PHP файлі. Наприклад, «шапка» в одному файлі, «меню» в іншому і т. д. Вся інформація, яка представлена на web-сайті розміщена в базі даних. Файли PHP зв'язуються з базою даних і отримують дану інформацію з неї.

Інформаційна карта веб-сайту: головна, послуги, товари, контакти.

Веб-сайт призначений для:

1) поширення інформації про послуги та товари;

2) надання інформації про магазин: обід, купівля, доставка, послуги;

3) інформувати про контакти магазину.

За допомогою сучасних Інтернет технологій мови програмування PHP та системи управління базами даних MySQL, було розроблено інформативний Web-сайт Інтернет-магазин комп'ютерної техніки, функціональна структура якого складається з Web-сайту та бази даних, дані елементи взаємодіють між собою.

Структура Web-сайту базується на адміністративній та користувальницкій частині.

Система управління контентом (адміністративна частина сайту) надає можливість додавання послуги, зміна послуги, видалення послуги, додавання товару, зміна товару та видалення товару.

У подальшому можливе розширення функцій даного web-сайту.

Враховуючи вище сказане, можна зробити висновок, що всі поставленні завдання в дипломній роботі виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Микитенко, Я., & Делембовський, М. (2020). БЕЗДРОТОВА МЕРЕЖА Wi-Fi НАВЧАЛЬНОГО КОРПУСУ. Матеріали конференцій Молодіжної наукової ліги.

2. Fontes, R. D. R., & Rothenberg, C. E. (2016, August). Mininet-wifi: A platform for hybrid physical-virtual software-defined wireless networking research. In Proceedings of the 2016 ACM SIGCOMM Conference (pp. 607-608).

3. Hernandez, L., Balmaceda, N., Hernandez, H., Vargas, C., De La Hoz, E., Orellano, N., ... & Uc-Rios, C. E. (2019, July). Optimization of a wifi wireless network that maximizes the level of satisfaction of users and allows the use of new technological trends in higher education institutions. In International Conference on Human-Computer Interaction (pp. 144-160). Springer, Cham.

4. Redondi, A. E., Cesana, M., Weibel, D. M., & Fitzgerald, E. (2016, September). Understanding the WiFi usage of university students. In 2016 International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 44-49). IEEE.

5. Sur, S., Pefkianakis, I., Zhang, X., & Kim, K. H. (2017, October). WiFi-assisted 60 GHz wireless networks. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (pp. 28-41).

6. Bhatt, A., & Patoliya, J. (2016, February). Cost effective digitization of home appliances for home automation with low-power WiFi devices. In 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB) (pp. 643-648). IEEE.

7. Yan, C., Xu, Y., Shen, J., & Chen, J. (2016, October). A combination of VLC and WiFi based indoor wireless access network and its handover strategy. In 2016 IEEE International Conference (ICUWB) (pp. 1-4). IEEE.

8. Shi, J., Meng, L., Striegel, A., Qiao, C., Koutsonikolas, D., & Challen, G. (2016, April). A walk on the client side: Monitoring enterprise wifi networks using

smartphone channel scans. In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications (pp. 1-9). IEEE.

9. Zheng, L., Chen, J., Joe-Wong, C., Chee, W. T., & Chiang, M. (2017, May). An economic analysis of wireless network infrastructure sharing. In 2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt) (pp. 1-8). IEEE.

10. Khanchuea, K., & Siripokarpirom, R. (2019, March). A multi-protocol IoT gateway and WiFi/BLE sensor nodes for smart home and building automation: Design and Implementation. In 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES) (pp. 1-6). IEEE.

11. Han, K., Li, S., Tang, S., Huang, H., Zhao, S., Fu, G., & Zhu, Z. (2018). Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing. IEEE Access, 6, 26567-26577.

12. Min, C., & Jinhao, Z. (2021, March). The Application of WiFi 6 Technology in Underground Mine. In IOP Conference Series: Earth and Environmental Science (Vol. 687, No. 1, p. 012153). IOP Publishing.

13. Guillen-Perez, A., Sanchez-Iborra, R., Cano, M. D., Sanchez-Aarnoutse, J. C., & Garcia-Haro, J. (2016, November). WiFi networks on drones. In 2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT) (pp. 1-8). IEEE.

14. Omar, H. A., Abboud, K., Cheng, N., Malekshan, K. R., Gamage, A. T., & Zhuang, W. (2016). A survey on high efficiency wireless local area networks: Next generation WiFi. IEEE Communications Surveys & Tutorials, 18(4), 2315-2344.

15. Luo, J., Zhang, Z., Liu, C., & Luo, H. (2018). Reliable and cooperative target tracking based on WSN and WiFi in indoor wireless networks. IEEE Access, 6, 24846-24855.

16. Balzano, W., Murano, A., & Vitale, F. (2018). SNOT-WiFi: sensor network-optimized training for wireless fingerprinting. Journal of High Speed Networks, 24(1), 79-87.

17. Xu, C., Jin, W., Zhao, G., Tianfield, H., Yu, S., & Qu, Y. (2017). A novel multipath-transmission supported software defined wireless network architecture. IEEE access, 5, 2111-2125.

18. Yang, L., Zhang, W., Zhang, Y., & Zhang, J. (2019, April). Hybrid Optical Wireless Network Based on Visible Light Communications (VLC)-WiFi Heterogeneous Interconnection. In 2019 2nd International Conference on Communication Engineering and Technology (ICCET) (pp. 35-38). IEEE.

19. Ren, Z., Chen, C., & Zhang, L. (2018, March). Security protection under the environment of WiFi. In 2017 International Conference Advanced Engineering and Technology Research (AETR 2017) (pp. 50-54). Atlantis Press.

20. Brincat, A. A., Lombardo, A., Morabito, G., & Quattropani, S. (2019). On the use of Blockchain technologies in WiFi networks. Computer Networks, 162, 106855.

21. Van de Belt, J., Ahmadi, H., & Doyle, L. E. (2017). Defining and surveying wireless link virtualization and wireless network virtualization. IEEE Communications Surveys & Tutorials, 19(3), 1603-1627.

22. Andrews, J. G., Zhang, X., Durgin, G. D., & Gupta, A. K. (2016). Are we approaching the fundamental limits of wireless network densification?. IEEE Communications Magazine, 54(10), 184-190.

23. Wu, Q., & Zhang, R. (2019). Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. IEEE Transactions on

Wireless Communications, 18(11), 5394-5409.

24. Gehlot, A., & Rajavat, A. (2016, March). Handoff between WiMAX and WiFi wireless networks. In 2016 Symposium on Colossal Data Analysis and Networking (CDAN) (pp. 1-5). IEEE.

25. Zou, L., Javed, A., & Muntean, G. M. (2017, June). Smart mobile device power consumption measurement for video streaming in wireless environments: WiFi vs. LTE. In 2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) (pp. 1-6). IEEE.

26. Park, P., Ergen, S. C., Fischione, C., Lu, C., & Johansson, K. H. (2017). Wireless network design for control systems: A survey. IEEE Communications Surveys & Tutorials, 20(2), 978-1013.

27. Ertam, F., Kilincer, I. F., Yaman, O., & Sengur, A. (2020, September). A New IoT Application for Dynamic WiFi based Wireless Sensor Network. In 2020 International Conference on Electrical Engineering (ICEE) (pp. 1-4). IEEE.

28. Kosek-Szott, K., Gozdecki, J., Loziak, K., Natkaniec, M., Prasnal, L., Szott, S., & Wagrowski, M. (2017). Coexistence issues in future WiFi networks. IEEE Network, 31(4), 86-95.

29. Yuan, C., Fei, L., Jianxin, C., & Wei, J. (2016, May). A smart parking system using WiFi and wireless sensor network. In 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) (pp. 1-2). IEEE.

30. Wu, Q., & Zhang, R. (2019). Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts. IEEE Transactions on Communications, 68(3), 1838-1851.

31. Wei, X., Wang, Q., Wang, T., & Fan, J. (2016). Jammer localization in multi-hop wireless network: a comprehensive survey. IEEE Communications Surveys & Tutorials, 19(2), 765-799.

32. Sarra, E., & Ezzedine, T. (2016, September). Performance improvement of the

wireless body area network (WBAN) under interferences. In 2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom) (pp. 1-6). IEEE.

33. Wang, J., Jiang, C., Han, Z., Ren, Y., & Hanzo, L. (2016). Network association strategies for an energy harvesting aided super-WiFi network relying on measured solar activity. IEEE Journal on Selected Areas in Communications, 34(12), 3785-3797.

34. Wang, S., Li, B., Yang, M., & Yan, Z. (2018, October). Intrusion detection for WiFi network: A deep learning approach. In International Wireless Internet Conference (pp. 95-104). Springer, Cham.

35. Tran, M. A. T., Le, T. N., & Vo, T. P. (2018, November). Smart-config wifi technology using ESP8266 for low-cost wireless sensor networks. In 2018 International Conference on Advanced Computing and Applications (ACOMP) (pp. 22-28). IEEE.

36. Dey, K. C., Rayamajhi, A., Chowdhury, M., Bhavsar, P., & Martin, J. (2016). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network–Performance evaluation. Transportation Research Part C: Emerging Technologies, 68, 168-184.

37. Wu, X., Soltani, M. D., Zhou, L., Safari, M., & Haas, H. (2021). Hybrid LiFi and WiFi networks: A survey. IEEE Communications Surveys & Tutorials.

38. Richart, M., Baliosian, J., Serrati, J., Gorricho, J. L., Agüero, R., & Agoulmine, N. (2017, November). Resource allocation for network slicing in WiFi access points. In 2017 13th International conference on network and service management (CNSM) (pp. 1-4). IEEE.

39. Wen, Y., & Liu, T. (2018, October). WIFI Security Certification through Device Information. In 2018 International Conference on Sensor Networks and Signal Processing (SNSP) (pp. 302-305). IEEE.