

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ»**

**Кафедра Економічної кібернетики та інженерії програмного
забезпечення**

ДО ЗАХИСТУ ДОПУЩЕНА
Зав.кафедри _____
д.е.н., доцент Левицький С.І.

**БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА
РОЗРОБКА ВЕБ-САЙТУ СЕРВІСНОГО ЦЕНТРУ
З ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ**

Виконав
ст. гр. ІПЗ-118

(підпис)

Д.В. Скляр

Керівник
ст. викладач

(підпис)

Л.М. Шокотько

Запоріжжя
2022

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра Економічної кібернетики та інженерії програмного забезпечення

ДО ЗАХИСТУ ДОПУЩЕНА
Зав.кафедри _____
д.е.н., доцент Левицький С.І.

1.1 ЗАВДАННЯ
1.1.1 НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Скляр Дмитру Вячеславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка web-сайту сервісного центру з обслуговування комп'ютерних систем

керівник роботи Шокотько Людмила Миколаївна, ст. викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом вищого навчального закладу № 06.-50 від 18.02.2022 р.

2. Строк подання студентом роботи 17.06.2022 р.

3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- аналіз предметної області;
- визначити вимоги до функціоналу та базових компонент;
- розглянути програмно-технологічні платформи для створення сайту;
- технології створення сайту і вибір майданчика для нього;
- проектування структури сайту та сховища даних;
- складання технічного завдання та схеми головної сторінки;
- реалізація архітектури сайту;
- тестування, seo-оптимізація сайту
- зробити висновки за результатами виконаної роботи.

5. Дата видачі завдання 17.01.2022 р.

Студент

(підпис)

Скляр Д. В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Шокотько Л. М.

(прізвище та ініціали)

РЕФЕРАТ

Бакалаврська дипломна робота: 58 сторінок, 23 рисунка, 3 таблиці,
22 джерела інформації.

Об'єкт дослідження: web-сайти сервісних центрів.

Предметом дослідження є сукупність теоретичних, методичних і практичних питань створення ефективного сайту сервісного центру з обслуговування комп'ютерних систем.

Мета дослідження: теоретичне обґрунтування і дослідження методології створення web-сайту сервісного центру з обслуговування комп'ютерних систем.

Для реалізації задачі було створено web-сайт, розроблено базу даних, призначену для зберігання вхідної інформації. Відповідно до розробленої моделі, був визначений алгоритм побудови програми. Для реалізації алгоритму задачі було обрано мову програмування PHP, з використанням мови розмітки HTML 5 та локальне серверне середовище MAMP.

Результатом кваліфікаційної роботи стала розробка web-сайту та бази даних, які вирішують поставлену задачу. Програмний комплекс, розроблений на мові PHP, з використанням мови розмітки HTML 5, яка призначена для створення користувацьких інтерфейсів web-сайту.

Результати роботи є універсальними і можуть бути застосовані в будь-яких сервісних центрах.

БАЗА ДАНИХ, ВЕБ-САЙТ, КОНСОЛІДАЦІЯ ІНФОРМАЦІЇ,
КОНТЕНТ, СЕРВЕР, СЕРВІСНИЙ ЦЕНТР, HTML, PHP

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	6
ВСТУП	7

РОЗДІЛ 1	ХАРАКТЕРИСТИКА ТА ЗАГАЛЬНИЙ ОГЛЯД	
	СУЧАСНИХ ІНТЕРНЕТ-ТЕХНОЛОГІЙ В КОНТЕКСТІ	
	КОМЕРЦІЙНОЇ ДІЯЛЬНОСТІ	10
1.1	Сайт як інструмент online-просування послуг в мережі	10
1.2	Адаптивний дизайн і етапи розробки web-сайту	11
1.3	Інструментарій для поетапної побудови сайту	14
РОЗДІЛ 2	ПРОЕКТУВАННЯ WEB-САЙТУ СЕРВІСНОГО	
	ЦЕНТРУ З ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ	19
2.1	Проектування структури сайту	19
2.2	Проектування сховища даних	22
2.3	Створення системи управління web-ресурсом	25
РОЗДІЛ 3	РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ WEB-САЙТУ	
	СЕРВІСНОГО ЦЕНТРУ З ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ	
	СИСТЕМ	29
3.1	Програмна реалізація проекту	29
3.2	Функціональне та юзабіліті-тестування	44
3.3	Просування сайту	51
ВИСНОВКИ	54
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
ТА ТЕРМІНІВ

Слово / словосполучення	Скорочення	Умови використання
I		

Слово / словосполучення	Скорочення	Умови використання
інформаційні технології	ІТ	
О		
об'єктно-орієнтоване програмування	ООП	
П		
програмне забезпечення	ПЗ	-
С		
система управління базами даних	СУБД	
А		
Asynchronous Javascript and XML	AJAX	
С		
Cascading Style Sheets 3	CSS3	
Cross-Site Request Forgery	CSRF	
D		
Dynamic HTML	DHTML	-
Domain name system	DNS	
H		
Hyper Text Markup Language	HTML	
P		
Personal Home Page Tools	PHP	
R		
Really Simple Syndication	RSS	-
S		
Secure Shell	SSH	-
SSH File Transfer Protocol	SFTP	-
U		
Uniform Resource Locator	URL	
User Experience	UX	
User Interface	UI	
W		
Virtual Private Server	VPS	-
World Wide Web	WWW	-

ВСТУП

У зв'язку з введенням воєнного стану в Україні, нагальним стає питання організації роботи сервісних центрів, створення платформ для надання інформації, консультацій та отримання замовлень на певні послуги з обслуговування комп'ютерних систем.

Завдання, які реалізуються компаніями за допомогою Інтернету, умовно можна розбити на кілька типів:

- 1) реклама та створення іміджу організації;
- 2) маркетинг товарів;
- 3) здійснення продажів за допомогою Інтернету;
- 4) надання консультаційної допомоги споживачам;
- 5) організація каналу мовлення;
- 6) засоби масової інформації (ЗМІ).

Будь-який бізнес, який здійснюється за допомогою взаємодії з широкою аудиторією, вимагає постійної реклами. Кілька років тому до звичайних засобів реклами додався новий, і часом більш ефективний інструмент.

Починаючи з середини 90-х років, у всьому світі спостерігається зростання активності в галузі електронної торгівлі. Слідом за великими компаніями, що виробляють комп'ютерне обладнання в Інтернет стали виходити торговці традиційними товарами. З'явилася велика кількість книгарень, магазини компакт-дисків і відеокасет, винні магазини. Зараз практично будь-які товари можна купити через Інтернет.

Представляючи інформацію про себе і свої продукти, фірма не тільки привертає потенційних покупців, а й отримує можливість зворотного зв'язку з ними. Крім цього, в Інтернеті простіше організувати адресну рекламу, коли web-сайт рекламується серед людей, які можуть бути потенційними покупцями.

Довгий час символом презентабельності фірми була наявність web-сайту, що само по собі вважалось показником її гнучкості та хорошої здатності адаптуватися до умов мінливого бізнесу.

В даний час багато фірм, не шкодуючи коштів, розвивають свої сайти для додання кращого вигляду "особі" фірми у Всесвітній павутині. web-сайти надають унікальну можливість для фірми ненав'язливо представити великий обсяг інформації про себе, вироблених продуктах або послугах, що надаються. З іншого боку, за інтересом до того чи іншого товару, зміни рейтингу відвідувань сайту можна судити про ефективність проведених рекламних акцій. Багато фірм, при випуску нових продуктів, спеціально створюють сайт з їх детальним описом, а потім через цей сайт отримують відгуки покупців і оперативно корегують свою діяльність щодо цих виробів.

Окрім представлення опису товару, за допомогою Інтернету, можна здійснювати його продаж. Вже багато компаній мають свої електронні магазини.

Сайтів в Інтернеті мільйони. Призначення сайтів найрізноманітніше. До них відносяться бізнес-сайти, інформаційні, розважальні, рекламні, навігаційні, художні, персональні, комбіновані сайти. Особливу категорію сайтів складають сайти для бізнесу (бізнес-сайти, комерційні сайти), які створюються для вирішення конкретних бізнес-завдань. Під бізнес-сайтом мається на увазі сайт, призначений для отримання прибутку його власником за рахунок залучення потенційних клієнтів з конкретної цільової аудиторії. Основна аудиторія бізнес-сайту - потенційні і справжні клієнти фірми або організації. Додаткова аудиторія часто включає в себе потенційних і дійсних інвесторів, а також зацікавлені організації, такі як засоби інформації і фірми-конкуренти.

Основна мета будь-якого комерційного сайту - обслуговування користувачів таким способом, який приносить компанії вигоду або

безпосередньо, або опосередковано. Мета будь-якого бізнесу - отримання прибутку.

Доцільність створення того чи іншого типу сайту для бізнесу визначає коло вирішуваних сайтом завдань, де необхідно вибрати основну мету створення сайту, другорядні цілі, бюджет сайту.

Залежно від бюджету може змінюватись вибір основної форми документа, на основі якого буде розроблятися сайт. Від бюджету безпосередньо залежать: об'єм сайту, рівень його технічної складності, професійний рівень web-дизайнера, терміни створення сайту, число відвідувачів сайту і, отже, прибуток сайту[1].

Успішне ведення бізнесу неможливе без наявності професійно створеного корпоративного сайту.

Тому тема бакалаврської дипломної роботи є актуальною та обумовлює її мету: проектування і розробка web-сайту сервісного центру з обслуговування комп'ютерних систем.

Об'єкт дослідження: web-сайти сервісних центрів.

Предметом дослідження є сукупність теоретичних, методичних і практичних питань створення ефективного сайту сервісного центру з обслуговування комп'ютерних систем.

Практична значимість роботи: розроблений сайт є універсальним і може бути застосованим в будь-яких сервісних центрах.

Відповідно до завдань дослідження було визначено структуру роботи. Вона складається з вступу, трьох розділів, висновків, переліку використаних джерел.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА ЗАГАЛЬНИЙ ОГЛЯД СУЧАСНИХ ІНТЕРНЕТ-ТЕХНОЛОГІЙ В КОНТЕКСТІ КОМЕРЦІЙНОЇ ДІЯЛЬНОСТІ

1.1. Сайт як інструмент online-просування послуг в мережі

В даний час створення сайтів набирає велику популярність. Стали звичними випадки, коли корпоративний сайт створюється ще до офіційного відкриття самої компанії і заздалегідь вкладаються гроші в просування даного ресурсу. Зростання популярності сайтів пояснюється тим, що більшість людей вважають за краще шукати необхідну інформацію на просторах Інтернету, причому не важлива область: починаючи від миття машин і покупки комп'ютерних комплектуючих, закінчуючи установкою котельного обладнання, будівництва будинків і квартир - всю цю інформацію шукають саме в Інтернеті.

Переваги створення сайту:

- 1) найдешевший спосіб реклами: Створення самого сайту може здатися відносно дорогим (в середньому якісний сайт коштує від 15 тис. до 30 тис. гривень), та й просування сайту вимагає витрат. Але на відміну від інших видів реклами - сайт не вимагає постійних пропорційних вкладень. Ви можете зробити якісний сайт, просунути його і обмежуватися вкладенням в 1-2 тис. гривень на місяць на підтримку позицій. Вам не доведеться витратити великих сум, як у випадку з рекламою по телевізору або радіо;
- 2) цільова аудиторія. Сайт здатний залучити цільову аудиторію - тобто саме тих людей, які шукають послуги, набираючи їх в пошукових системах і потрапляючи на ваш сайт. Таке співвідношення зацікавлених клієнтів не отримати ні в жодному іншому вигляді реклами. Тому віддача буде вище;

- 3) сайт - це об'ємний ресурс, де можна успішно презентувати товар, викласти повний перелік товарів, описати всі переваги вашої компанії, контакти і т.д. Тобто при правильній структурі сайту відвідувач отримає потрібний обсяг інформації, що набагато перевершує той, що він зможе отримати з флаєра або рекламного ролика;
- 4) функціональність. Сайт дарує безліч нових можливостей: ви можете уявити весь перелік послуг з реалізацією записи онлайн в зручний час (що заощадить час ваших фахівців, яким не потрібно буде приймати заявки по телефону, а також зробить процедуру зручніше для клієнтів). Можна викласти повний каталог товарів з можливістю замовлення онлайн. Це конкурентна перевага, яка допоможе отримати безліч додаткових замовлень, так як в наш час багато клієнтів не люблять витратити свій час на поїздки в магазини. Всі ці функції не можуть бути реалізовані іншими способами реклами;
- 5) презентабельність. Компанія без сайту - це нонсенс в сучасному світі.

Виходячи з усіх переваг, можна зробити висновок, що інтернет - це найкращий вид реклами.

1.2. Адаптивний дизайн і етапи розробки web-сайту

Структура сайту - це логічна побудова всіх сторінок ресурсу. Схема, за якою розподіляється шлях до папок, категорій, підкатегорій, карток товарів (якщо вони передбачені). З технічної точки зору, навігація ресурсу являє собою набір URL, логічно побудованих в певній послідовності. Структура взаємопов'язана з семантичним ядром. Саме воно говорить про те, які папки і документи повинні бути присутніми на сайті. Тому, зібравши семантику, вже можна зробити начерки схеми побудови кожного майбутнього урла.

Основною функціональною вимогою сайту такої спеціалізації є можливість складати і відправляти заявки на надання послуг. Користувач повинен мати можливість переглянути перелік наданих послуг та їх прайс-лист, після чого зробити заявку і зв'язатися з оператором, який передасть сформульовану заявку з зауваженнями працівникам сервісного центру. Система повинна зберігати свою працездатність при зверненні користувачів до неї через різні Internet браузері. Інформація повинна відображатися повністю та коректно.

Інформаційні вимоги:

- 1) сторінка, що містить коротку інформацію про призначення сайту і новини, вона ж головна.
- 2) сторінка для створення і відправки форми користувачами.
- 3) сторінка з контактною інформацією фірми.

Ми вже знаємо, що логічна і проста структура швидше і легше проглядається пошуковим алгоритмом, що призводить до гарних результатів з точки зору індексації. Щоб сторінка простіше ранжувалась і виходила в топ по певних запитах, необхідно опрацювати ЛЗУ. "Людино зрозумілі" URL допоможуть користувачеві зорієнтуватися на сторінці.

Приклад неправильного URL:

`mysite.com.ua/index.php?docid=17_88UaWp8hXtvMnFe8Vj8MQItEPbk&ln=ru`

Приклад правильного: `mysite.com.ua/catalog/divany/uglovooy`

Як бачите, вкладеність каталогу проглядається тільки в другому випадку, що дозволяє і користувачеві, і пошуковику прочитати і подивитися URL, зрозумівши, що знаходиться на сторінці.

Основні етапи створення web-сайту:

- 1) постановка цілей і завдань сайту;
- 2) створення, опрацювання технічного завдання (ТЗ) на розробку сайту;
- 3) прототипування;

- 4) створення макета дизайну сайту;
- 5) верстка;
- 6) програмування;
- 7) наповнення контентом;
- 8) тестування;
- 9) здача готового проекту клієнтові.

Будь-яка структура сайту незалежно від його виду нагадує деревоподібну схему, де головна сторінка - це стовбур, а категорії і підкатегорії - гілки і листя. І якщо головна - це основа, без неї не обійтися в будь-якому сайті, то принцип розміщення і побудови категорій і підкатегорій залежить від виду сайту і його цілей.

При розробці сайту повинні бути використані переважно світлі і контрастні кольорові рішення:

- 1) оформлення має бути розроблено в досить консервативному ключі;
- 2) основні розділи сайту повинні бути доступні з першої сторінки.

Як було зазначено вище, кількість прикладних програм в Інтернет стрімко збільшуються і особливо актуальні проектні рішення з використанням БД, що дозволяють вирішувати багатьох завдань.

З огляду на велику кількість ресурсів наявних в інтернеті, сучасні web-вузли як правило використовують мережеві СУБД. Застосування таких СУБД дозволяє швидко і ефективно обробляти запити користувача і надійно зберігати дані. У дипломній роботі розробляється web-вузол на прикладі бази даних по «електронній комерції». На даний час Інтернет магазини є найпоширенішою послугою в мережі Інтернет.

Web-сайт – це невід’ємна частина сучасної інтернет фірми. У будь-якого web-сайту є певна структура, всі хороші інтернет-ресурси повинні бути виконані у відповідності з правилами будівництва сайту. URL сайту повинні відповідати правилам людинозрозумілих URL (ЛЗУ).

1.3. Інструментарій для поетапної побудови сайту

Два найпопулярніших на даний момент web-сервера, це Apache - вільно розповсюджується від "Apache Software Foundation" і IIS (Internet Information Services) від фірми Microsoft, що входить до складу ОС Windows (27% і 40% від усіх сайтів, відповідно). Функціонально обидва сервера практично ідентичні.

І, хоча обидва сервера мають можливість підключення програм практично на будь-якій мові програмування, найбільшої зручності в розробці та інтеграції можна досягти використовуючи:

- 1) разом з Apache: Perl, Python, PHP;
- 2) разом з IIS: ASP, ASP.NET.

Значить, вибір web-сервера зав'язаний на вибір платформи для розробки програмного коду.

Існує багато мов програмування, але потрібно вибрати одну, для цього буде приведена коротка класифікація популярних мов програмування. З них потрібно виділити плюси для дипломного завдання та оцінити їх функціонал.

Позитивні характеристики мови Java:

- 1) це одна з найпопулярніших мов програмування, що використовується понад 20 років;
- 2) мова Java є універсальною завдяки віртуальній машині Java (Java Virtual Machine, JVM). Вона дозволяє коду на Java однаково працювати на всіх сумісних платформах. JVM - своєрідна прошарок, в якій Java-програма перетвориться в код, який може виконуватися на будь-якій машині;
- 3) незважаючи на те, що Java надзвичайно популярна серед розробників ПЗ, вона складніша для новачка, ніж, скажімо, Python. Проте, у Java величезне співтовариство, яке дасть відповідь практично на будь-яке питання новачка або професіонала;

Позитивні характеристики PHP:

- 1) на PHP працює близько 78.2% всіх web-сайтів. Мова вперше була представлена в 1995 році, коли для створення динамічних сайтів існувало не так багато можливостей;
- 2) оскільки це динамічно-типізована мова, для однієї проблеми можна знайти відразу кілька рішень. Це одночасно означає, що одна і та ж ділянка коду може вести себе по-різному в залежності від конкретної ситуації, що робить програми на PHP складно масштабованими і в деяких випадках повільними;
- 3) PHP допускає помилки. Так, програма, в кодї якої є проблеми, буде виконуватися поки не досягне ділянки з помилкою;
- 4) у мови дуже велика спільнота, причому для новачків є велика кількість навчальних матеріалів. Мова постійно оновлюється.

Позитивні характеристики .NET:

- 1) ASP.NET - це відповідь Microsoft на Java. .NET являє собою фреймворк, який використовується для створення сайтів на таких мовах, як Visual Basic (VB), C #, F # та інші;
- 2) основа мови - архітектурний шаблон MVC (Model-View-Controller). У цій схемі контролер приймає запити користувача і взаємодіє з моделлю для обробки даних. Потім результат вже передається в відображення, з'являючись у вигляді інтерфейсу web-сторінки.

Розглянемо найпопулярніші мови програмування для кожного web-сервера.

Мова програмування PHP - це мультипарадигменна сценарна інтерпретована мова програмування, застосовувана в основному на серверах з Apache і Nginx. Основні плюси: C-подібний синтаксис, вбудовування безпосередньо в HTML файл, розширюваність модулями, широка поширеність в великих проектах.

Однак, є і недоліки: інтерпретованість, для складних проектів необхідність в додаткових модулях і необхідність в пошуку стороннього інтегрованого середовища розробки (при збереженні можливості писати код в простий текстовий файл).

ASP.NET - це web-платформа, що надає всі необхідні служби для створення web-додатків. ASP.NET створена на основі платформи .NET Framework, тому всі функції .NET Framework доступні для додатків ASP.NET. Додатки можуть бути написані на будь-якій мові, сумісному з середовищем CLR, включаючи Visual Basic і C#.

.NET в своїй основі є загальномовним середовищем виконання (CLR), а значить володіє динамічною компіляцією і властивостями всіх використовуваних мов програмування.

До основних плюсів ASP.NET можна віднести: Динамічну компіляцію, наявність власного середовища розробки, великий набір модулів, що входять до складу середовища розробки, можливість використання популярних мов прикладного програмування (C #, наприклад).

Виходячи з характеристик різних технологій зручною, ефективною, а також безкоштовною була обрана технологія PHP. До конкретних завдань відносяться:

- 1) розробка структури web-вузла;
- 2) вибір програмного інструментарію;
- 3) розробка зручного і зрозумілого для користувача інтерфейсу;
- 4) розробка структури і форм запитів;
- 5) розробка оптимальної навігаційної структури системи (web-вузла);
- 6) розроблений web-проект повинен бути оптимізований під використанням в реальній мережі (мати мінімально можливий обсяг запитуваних web-сторінок).

До загальних вимог слід віднести те, що обсяг інтерактивних і графічних елементів повинен бути мінімізований. Інтернет магазин повинен

мати систему захисту від несанкціонованого доступу. Система повинна бути орієнтована на подальший розвиток.

Основною вимогою до розробки інформаційного забезпечення є вибір СУБД (Система Управління Базою Даних) для зберігання інформації про товари. В якості такої СУБД була обрана MySQL, оскільки дана СУБД має клієнт - серверну архітектуру, функціонує в операційних системах сімейства Unix, займає не дуже багато машинної пам'яті. Підтримка сервера MySQL автоматично включається в поставку PHP.

Вимоги до інтерфейсу інформаційної системи:

- 1) для інтерактивної обробки інформації система повинна включати форму для введення даних;
- 2) колірну палітру інтерфейсів витримати в стилі Windows. Дозвіл екрану: 1280 * 1024 - великий шрифт;
- 3) інтерфейс повинен бути захищеним від неправильних дій користувача і забезпечений системою підказок.
- 4) Вимоги до технічного забезпечення:
- 5) вимоги до сервера - персональний комп'ютер стандартної конфігурації, вільне місце на жорсткому диску 1 Гб;
- 6) вимоги до клієнта - відповідно до рекомендованих вимог операційної системи.

Вимоги до програмного забезпечення:

- 1) на стороні сервера - операційна система Windows XP і пізніших версій, СУБД MySQL;
- 2) безкоштовне локальне серверне середовище MAMP дозволить запуснути сервер Apache на додаток до MySQL як сервер баз даних;
- 3) вимоги до клієнта – операційна система Windows XP, браузері – Google Chrome 90+, Opera 10+.

До недоліків можна віднести підвищену складність роботи, так як потрібне знання всього комплексу середовища розробки.

В якості мови розмітки і мови стилів найвигідніше використовувати найостанніші версії, так як вони володіють найбільшим функціоналом, тому:

- 1) Мова розмітки - HTML 5;
- 2) Таблиці стилів - CSS 3.

Оскільки середовище та web-сервер з Apache вже присутні, було вирішено писати сайт на чистій мові HTML 5 з великою домішкою PHP скриптів та деякими JS додатками.

.

РОЗДІЛ 2

ПРОЕКТУВАННЯ WEB-САЙТУ СЕРВІСНОГО ЦЕНТРУ З ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

2.1. Проектування структури сайту

Перед написанням логіки архітектури клієнт-сервер почнемо писати HTML складову сайту, тобто головну сторінку і все що потрібно знати користувачу до відправки повідомлення - прайс-лист, порядок користування сайтом та невеликий віджет з відгуками клієнтів, який можливо прив'язати до великих сайтів відгуків.

Сайт буде починатись з головної сторінки, на якій буде декілька кнопок для переходу на інші сторінки сайту. Першою кнопкою нехай буде блог і новини в сфері ІТ, який може вести адміністратор сайту за бажанням, або це буде посиланням на більш великий ІТ-блог, де користувачі зможуть дізнаватись про новини.

Дві інші кнопки будуть виступати посиланнями на прайс-лист сайту, де будуть розписані усі можливі послуги, що персонал сервісного центру може надати.

Оскільки головна сторінка в більшості використовує мову розмітки HTML, можна одразу будувати весь інтерфейс сторінки зверху вниз, використовуючи ключові команди мови CSS для кращої візуальної якості сторінки. Але дизайн повинен бути консервативним, тому ми будемо використовувати просту структуру сервісного центру і поступово заповнювати сторінку контентом.

Отже, на головній сторінці також повинен бути логотип сервісного центру та вся доступна інформація про надання послуг. В якості логотипу в цій дипломній роботі буде використовуватись логотип Google. Він розміщений горизонтально і зможе бути заміщений іншим логотипом у будь-

яку мить. На головну сторінку в якості дизайну також буде розміщено відео з постановкою системи охолодження процесору, що одразу наведе потенційного користувача на думку про спеціалізацію сайту. На верхній частині головної сторінки буде розміщено кнопку «Увійти» з посиланням на сторінку реєстрації та авторизації, яку було зроблено першою.

Коли верхня частина головної сторінки буде завершена, користувач перейде вниз, де будуть розміщені основні позиції сервісу та їх ціни. Це зроблено для того щоб потенційний користувач міг одразу оцінити цінову політику сервісу. Сюди ж буде встановлено декілька шаблонних картинок для кращої візуальної оформленості.

Після того як клієнт перевірить цінову політику, сервісний центр залишить блок інформації про порядок роботи, в якому будуть зазначені основні етапи роботи з технікою клієнта. Також в цьому блоці буде зазначено час оплати роботи.

Нижче потенційний клієнт зможе побачити блок відгуків про сервісний центр. Зробимо три вікна із відгуками, які будуть гортатися самостійно або за допомогою стрілок.

У футері залишимо фактичну адресу сервісного центру та гугл мапу з цим місцем. У кінці також залишимо можливу сторінку у соціальній мережі Facebook, партнерів та ще одну кнопку з посиланням на блог.

Наступним кроком буде налаштування системи реєстрації та авторизації до головної сторінки сайту. Для цього в уявному хедері буде кнопка «Увійти» з посиланням на сторінку `index.php`.

Головна сторінка спланована і час переходити до посилань з кнопок. Перша кнопка містить напис «Ціни на ІТ-послуги», тож вона є посиланням на прайс-лист. Для більш спланованого вигляду створена карта сайту, яка детально демонструє його структуру та макет (рис. 2.1.).

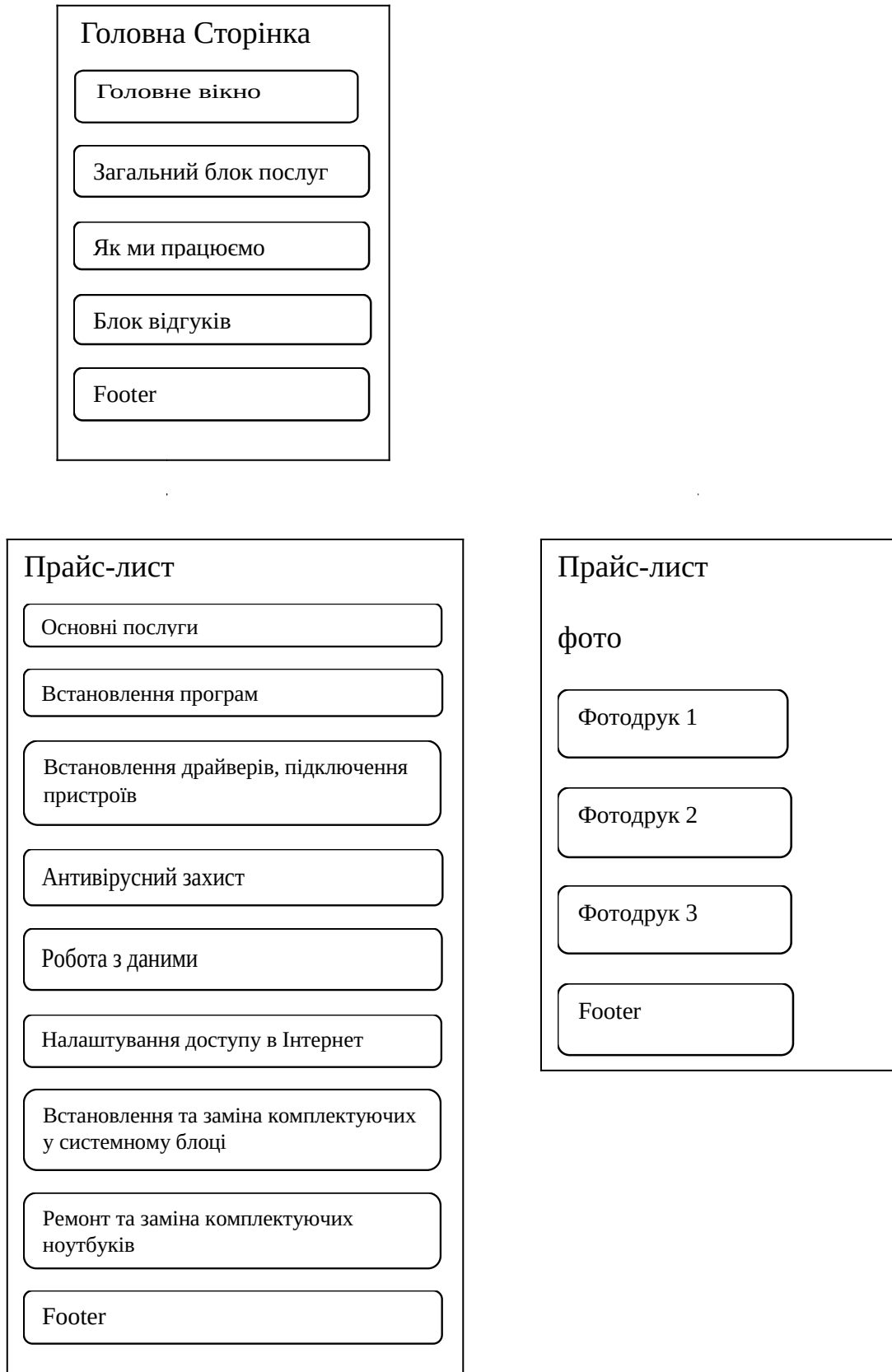


Рис. 2.1. Карта основних сторінок сайту

Сторінка прайс-листу буде простим переліченням послуг сервісного центру з зазначенням мінімальної ціни на послугу, буде виконана на мові та в стилі головної сторінки, міститиме у собі лише логотип сервісного центру зверху та дві кнопки. Одна кнопка буде функціональною з номером телефону, а інша буде повертати користувача на головну сторінку. За допомогою посилання з кнопки номеру телефону користувачі мобільних пристроїв зможуть миттєво подзвонити і уточнити всі питання. Для користувачів комп'ютерних пристроїв кнопка буде нефункціональною. У тому ж форматі зробимо іншу сторінку із цінами на роздруківку фотографій для більшої наповненості сайту.

Тепер сайт потребує системи реєстрації та авторизації, у подальшому вона буде виконана як сторінка з двома блоками: реєстраційним та авторизаційним. Ця сторінка має бути практично повністю зв'язана з обраною СУБД. Реєстраційний блок має запам'ятовувати користувача, в той час як авторизаційний блок має перевіряти наявність реєстрації.

2.2. Проектування сховища даних

В якості СУБД була обрана MySQL, а підтримка сервера MySQL автоматично включається в поставку PHP.

Обране безкоштовне локальне серверне середовище MAMP вже має в собі встановлений додаток PhpMyAdmin. Це web-додаток з відкритим кодом, написаний на мові PHP і представляє собою web-інтерфейс для адміністрування СУБД MySQL.

Модель роботи сайту така, що кожна сторінка представляє собою окремий модуль. Однак, функціонал деяких сторінок може повторюватися. Тому варто винести повторюваний функціонал в окремі модулі.

- 1) В результаті до модулів сторінок додадуться такі:
- 2) Модуль перевірки аутентифікації і прав;

3) Модуль роботи з надісланням повідомлення тільки для зареєстрованих користувачів;

4) Модуль роботи з базою даних.

Модуль роботи з надісланням повідомлення відображає основний функціонал сайту - формування заявки на основі відповіді на надані питання. Усі ці модулі будуть напряму з'єднані з СУБД і будуть обмінюватись з нею даними - додавати їх і перевіряти [2].

Схематичне створення цих модулів потребує зазначення деяких важливих рядків. По-перше у будь-якого запису завжди є `id` – він же порядковий номер запису, який повинен самостійно збільшуватись при реєстрації нових користувачів. Для цього існує спеціальна функція «`AUTO_INCREMENT`» (далі `A_I`), яка завжди збільшується при появі нових значень. Порядковий номер також не має бути від'ємним, для цього потрібно зазначити, що `id` приймає тільки невід'ємні значення, Тип даних «`unsigned int`» - беззнакове ціле число, як і тип `int` (знакове) займає в пам'яті 2 байти. Але на відміну від `int`, тип `unsigned int` може зберігати тільки позитивні цілі числа. Тому в усіх таблицях буде використовуватись цей тип даних. Поле `id` також буде мати помітку `Unique`, тобто буде ключовим унікальним полем таблиці.

Величини в даних типу `varchar` є рядками змінної довжини. Так само як і для стовпців `char`, можна задати стовпець `varchar` будь-якої довжини між 1 і 255. Однак, на противагу `char`, при зберіганні величин типу `varchar` використовується тільки необхідна кількість символів, з додаванням одного байт для запису довжини. Збережені величини пробілом не доповнюються, навпаки, кінцеві прогалини при зберіганні видаляються, саме тому в усіх рядках з використанням не тільки числового значення використовується `varchar`. Поле `id` передбачає лише цілі числові значення, тому типу `integer` буде достатньо. Також усі рядки окрім `id` та паролю повинні мати кодування «`utf8_general_ci`», тому що це кодування підтримує всі символи будь-якої

мови, але ігнорує регістр, тож заради безпеки пароль буде використовувати кодування «*utf8_bin*».

У таблиці 2.1 приведена база внутрішнього сховища реєстраційних даних – вона буде мати типові дані, а саме логін, ім'я та пароль користувача.

Таблиця 2.1

База даних внутрішнього сховища користувачів

№	Ім'я рядка	Тип даних	Кодування	Атрибут	Null	A_I
1	id	Int(11)	-	UNSIGNED	-	+
2	Login	Varchar(32)	<i>utf8_general_ci</i>	-	-	-
3	Password	Varchar(32)	<i>utf8_bin</i>	-	-	-
4	Name	Varchar(32)	<i>utf8_general_ci</i>	-	-	-

Другою таблицею буде база внутрішнього сховища повідомлень зареєстрованих клієнтів, потрібно лише замінити декілька рядків, схематично вона виконана це представлено на таблиці 2.2.

Таблиця 2.2

База даних внутрішнього сховища повідомлень

№	Ім'я рядка	Тип даних	Кодування	Атрибут	Null	A_I
1	id	Int(11)	-	UNSIGNED	-	+
2	Telephone	Varchar(32)	<i>utf8_general_ci</i>	-	-	-
3	Email	Varchar(32)	<i>utf8_general_ci</i>	-	-	-
4	Message	Varchar(9999)	<i>utf8_general_ci</i>	-	-	-

Таким чином маємо готову таблицю, яку треба буде перенести до SQL з дотриманням типізації даних [3].

2.3. Створення системи управління web-ресурсом

Оскільки управління web-ресурсом буде виконуватись за допомогою web-додатку PhpMyAdmin, необхідно створити декілька таблиць із даними, а саме:

Основну таблицю користувачів, де будуть зберігатись їх реєстраційні дані (рис. 2.2.). Файл Check.php буде використовувати код SQL для звернення до цієї таблиці:

```
$mysql = new mysqli('localhost','root','root','register-bd');
mysql->query("INSERT INTO `users` (`login`, `pass`, `name`)
VALUES ('$login', '$pass', '$name')");
mysql->close();
```

За допомогою команди «new mysqli» буде створюватись нова таблиця, до якої додадуться введені користувачем дані. Після взаємодії з базою даних обов'язково потрібно закрити взаємодію [4].

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A	J
id	INT	11	None		UNSIGNED	<input type="checkbox"/>	UNIQUE	id	<input checked="" type="checkbox"/>
login	VARCHAR	32	None	utf8_general_ci		<input type="checkbox"/>	---		<input type="checkbox"/>
pass	VARCHAR	32	None	utf8_general_ci		<input type="checkbox"/>	---		<input type="checkbox"/>
name	VARCHAR	16	None	utf8_general_ci		<input type="checkbox"/>	---		<input type="checkbox"/>

Table comments:

Collation: utf8_general_ci

Storage Engine: MyISAM

Рис. 2.2. Таблиця користувачів

А також таблицю для повідомлення для зареєстрованих користувачів, в якій вони будуть вводити свої телефон, електрону адресу та повідомлення про їх проблему (рис. 2.3.).

Structure						
Name	Type	Length/Values	Default	Collation	Attributes	Null A_I
id	INT	11	None		UNSIGNED	<input type="checkbox"/> <input checked="" type="checkbox"/>
tel	VARCHAR	225	None	utf8_general_ci		<input type="checkbox"/> <input type="checkbox"/>
email	VARCHAR	255	None	utf8_general_ci		<input type="checkbox"/> <input type="checkbox"/>
mass	VARCHAR	255	None	utf8_general_ci		<input type="checkbox"/> <input type="checkbox"/>

Рис. 2.3. Таблиця повідомлення

Таким чином система може за допомогою деяких команд мови SQL у PHP файлі сторінки додавати або перевіряти реєстрацію користувача. Для реалізації сторінки реєстрації та авторизації буде використовуватись файл `index.php`, який є головною сторінкою авторизації та реєстрації сервісу (рис. 2.4.). За допомогою рядка `<if ($_COOKIE['user'] ==>);»` ми будемо перевіряти авторизацію користувача. За допомогою коду таблиця зможе перевірити присутність користувача у системі:

```
if (count ($user)==0) {
    echo "Цей користувач не є дійсним";
    exit;
```

В іншому випадку користувач буде авторизований до системи за допомогою коду:

```
setcookie("user", $user['name'], time()+3600,"/");
```

Функція `<setcookie>` буде додавати користувача до системи на той період часу, який зазначено, у даному випадку на 3600 секунд. Для видалення користувача з системи використовується та ж функція, але атрибут буде вже `<-3600>`, після чого користувач повернеться до головної сторінки для незареєстрованих користувачів [5].

Форма реєстрації

Форма авторизації

Рис. 2.4. Сторінка реєстрації та авторизації

Отже, сторінка реєстрації зроблена, але їй потрібні допоміжні файли, що будуть додавати передану користувачем інформацію до СУБД, а також перевіряти реєстрацію. Для цього будуть використовуватись файли `Check.php` та `Auth.php`, які після виконання свого кода будуть переадресовувати користувача на головну сторінку авторизації після реєстрації та головну сторінку сайту для зареєстрованих користувачів після авторизації, відповідно.

Тепер, коли користувач може зареєструватись, авторизуватись та вийти, потрібно дати йому можливість зробити те, для чого він потрапив на сайт ремонту техніки. А саме залишити свої контактні дані і звернутись з проблемою, після чого персонал зв'яжеться з ним і допоможе її вирішити. В якості альтернатив на сайті також присутні телефон та адреса сервісного центру, але тут усе залежить від бажань адміністрації сайту.

Для відправки повідомлення будемо використовувати `Request.php`. Цей файл також буде прив'язаний до `checkmass.php`, який повторяє функції `Check.php`, але для таблиці повідомлення.

Загалом цей відрізок сайту має дещо складнішу структуру, ніж головна сторінка та її функціонал. Була створена карта авторизації сайту, де більш темні етапи є внутрішніми діями системи (рис. 2.5.). Після потрапляння на сторінку `Request.php` клієнт зможе написати свої дані та надіслати повідомлення.

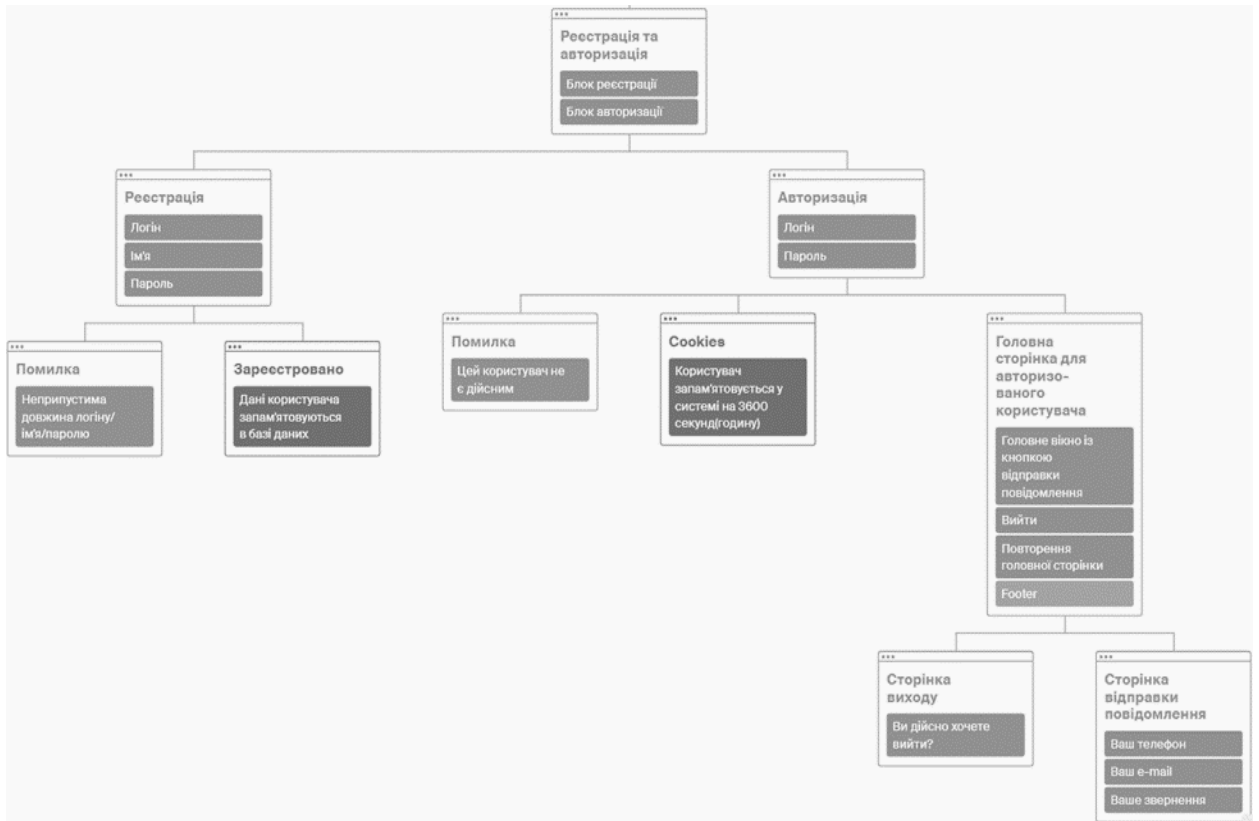


Рис. 2.5. Карта посилань сторінки реєстрації та авторизації

Таким чином бачимо що система має різні шляхи дій після очікуваних та помилкових дій користувача, а також має пояснення у випадку помилки [6].

РОЗДІЛ 3

РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ WEB-САЙТУ СЕРВІСНОГО ЦЕНТРУ З ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

3.1. Програмна реалізація проєкту

Розробку макету сайту розпочнемо з створення базової сторінки й одразу зробимо логотип в верхній частині сторінки, він буде вести на головну сторінку з усіх частин web-сайту [7]:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Сервісний центр Googledeep</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>
<body>
<div class="sow-slider-image-wrapper">
  <p><a href="index.html"></a></p>
  <p style="text-align: center"> Ремонт будь-яких видів
техніки </p>
  <p style="text-align: center"> Обслуговування техніки.
Налаштування програмної частини. Підтримка. </p>
</div>
</body>
</html>
```

Наступним кроком додаємо кнопку «увійти», яка буде переносити користувача на сторінку реєстрації та авторизації, яка була зроблена раніше:

```
<div class="button">
<p style="text-align: center">
<a href="index.php">
<button> Увійти</button>
</a> </div>
```

У верхню частину цієї сторінки, під логотип вставимо функціональний телефон компанії, який буде доступний користувачам мобільних пристроїв, а для користувачів комп'ютерів буде нефункціональний:

```
<p style="text-align: center">
<a style="color: #ffffff"
href="tel:+380930968531">+380930968531</a></p>
```

Завжди виконуємо центрування тексту та виконуємо шрифти, колір тексту за допомогою команд CSS:

```
<p style="text-align: center"><span style="font-family:
'Segoe UI', 'Microsoft JhengHei';text-align: center;font-weight:
400;letter-spacing: -1px;padding-bottom: 25px;font-size:
36px;color: #ffffff"><a style="color: #ffffff" href="tel:
+380930968531">+380930968531</a></span></p>
```

```
<p style="text-align: center"><span style="font-family:
'Segoe UI', 'Microsoft JhengHei';line-height: 1.19em;text-align:
center;font-weight: 400;letter-spacing: -1px;padding-bottom:
15px;font-size: 32px;color: #ffffff"> Ремонт будь-яких видів
техніки </span></p>
```

```
<p style="text-align: center"><span style="color:
#fcfcfc;font-size: 28px"><span style="font-family: 'Segoe UI',
'Microsoft JhengHei';line-height: 1.14286em;text-align:
center;font-weight: 300;letter-spacing: -1px;font-size: 24px">
Обслуговування техніки. Налаштування програмної частини.
Підтримка. </span></span></p>
```

Я використовую тег ``, щоб виділити частину інформації інших тегів та представити для неї свій стиль. Для різного тексту головної сторінки він добре впорається з командами CSS без необхідності працювати одночасно з двома файлами [8].

В початок елемента `body` також треба вставити колір заднього фону, аби білий текст було добре видно. В інтернеті досить часто зустрічаються колірні палітри із темним фоном:

```
<li class="sow-slider-image" style="background-color:
#565656">
</li>
```

Тепер у нас є 3 робочі посилання і декілька строк тексту на головній сторінці web-сайту (рис. 3.1.).

Також заради презентабельності головної сторінки можна поставити на фон відео. Будь-яке темне відео, приваблююче користувача. В цьому випадку візьмемо відео з установкою процесорного кулера у приближеному варіанті. Адміністратор сайту зможе в будь який час завантажити інше відео та замінити його. Для цього будемо використовувати функції автовідтворення, зациклення та мут, щоб відео запускалося при вході на сторінку, повторювалось та не відтворювало звук, відповідно:

```
<video class="sow-background-element" autoplay="" loop=""
muted="" playsinline="">
<source src="wp-content/uploads/2021/01/video.mp4"
type="video/mp4"></video>
</li>
```

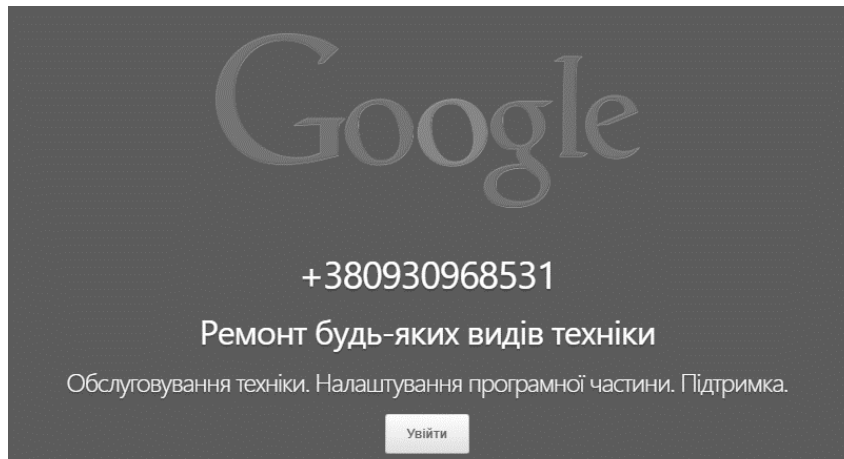



Рис. 3.1. Зовнішній вигляд елемента

Далі по шаблону сайту є три кнопки: блог та два прайс-листи:

```
<div class="sow-hero-buttons" style="text-align: center">
  <div class="so-widget-sow-button so-widget-sow-button-atom-
59014d9ecf3e">
    <div class="ow-button-base ow-button-align-center">
      <a href="https://БЛОГ.com/" class="ow-icon-placement-left
ow-button-hover" <span>
        <span class="sow-icon-fontawesome sow-fas" data-
sow-icon="&#xf35a;">
      </span> Блог з теми новин у сфері IT </span>
    </a></div></div>
```

Таким же чином зробимо ще 2 кнопки та стилізуємо їх. Вони будуть вести на інші сторінки сайту, до них можна буде повернутись потім. Тепер для закінчення першого вікна сайту треба додати останній напис та красиву лінію в кінці цього блока:

```
<h2 class='sow-headline'> Сервісний центр з ремонту техніки
Googledeep</h2>
  <div class="decoration">
    <div class="decoration-inside"></div></div>
```

В CSS клас декорації буде виконано так:

```
.decoration .decoration-inside {
  height: 1px;
  display: inline-block;
  border-top: 1px solid #eeeeee;
  width: 100%;
  max-width: 80%;
  margin-top: 0 !important;
  margin-bottom: 0 !important;
```

Верхня частина головної сторінки закінчена (Рис. 3.6.) і ми можемо переходити до наступних. Далі «Загальний блок послуг», який буде повторювати основні позиції прайс-листу, але буде оформлений на головній сторінці, тому що потенційний клієнт повинен у перші секунди перебування на web-сторінці магазину знайти основні товари.

Зробимо блок товарів у вигляді шести контейнерів із написом послуги, мінімальною ціною, уточнюючою картинкою послуги та більш детальними послугами по тематиці:

```
<div class="ow-pt-column" style="width: 100%">
  <div class="ow-pt-title">
    Усі види ремонту, налаштування, настройки ноутбуків </div>
  <div class="ow-pt-details">
    <div class="ow-pt-price">від 150 грн</div>
    <div class="ow-pt-per"> Виконуємо ремонт будь-яких ступенів
    складності</div></div>
  <div class="ow-pt-image">
    
  <div class="ow-pt-features">
    <div class="ow-pt-feature ow-pt-feature-even " title="">
```

```

<div class="sow-feature-text">
<p>Чистка ноутбука від 250грн</p></div></div>
<div class="ow-pt-feature ow-pt-feature-odd" title="">
<div class="sow-feature-text">
<p>Заміна SSD від 150грн</p></div></div>
<div class="ow-pt-feature ow-pt-feature-even" title="">
<div class="sow-feature-text">
<p>Заміна RAM від 100грн</p></div></div>
<div class="ow-pt-feature ow-pt-feature-odd" title="">
<div class="sow-feature-text">
<p>Полагодження клавіатури від 200грн</p></div></div>
<div class="ow-pt-feature ow-pt-feature-even" title="">
<div class="sow-feature-text">
<p>Заміна екрана від 250грн</p>
</div></div></div></div>

```

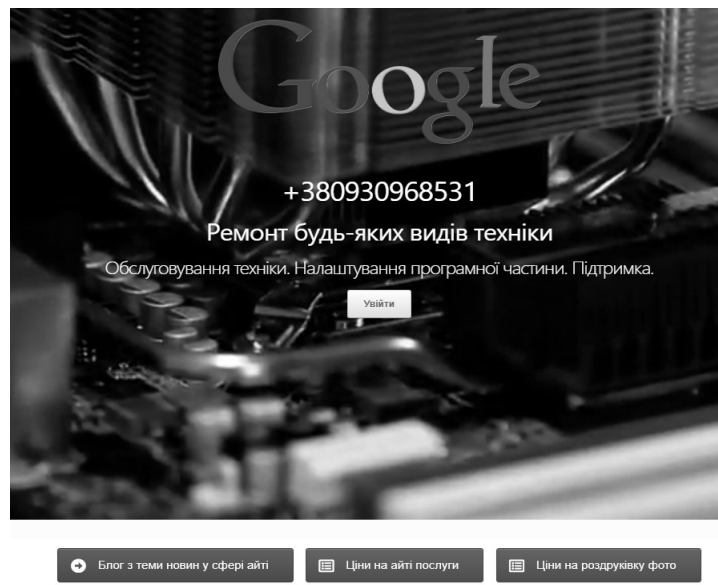


Рис. 3.2. Зовнішній вигляд головної сторінки

У тому ж вигляді зробимо ще 5 таких вікон із різними послугами та поставимо їх у два ряди.

За допомогою feature-odd та feature-even в CSS файлі зробимо красиву зебро образну стрічку під написами:

```
.ow-pt-feature-odd {
  background: #eeeeee;}
.ow-pt-feature-even {
  background: #f9f9f9;}
```

Цей блок тепер виглядає добре для клієнта (рис. 3.3):

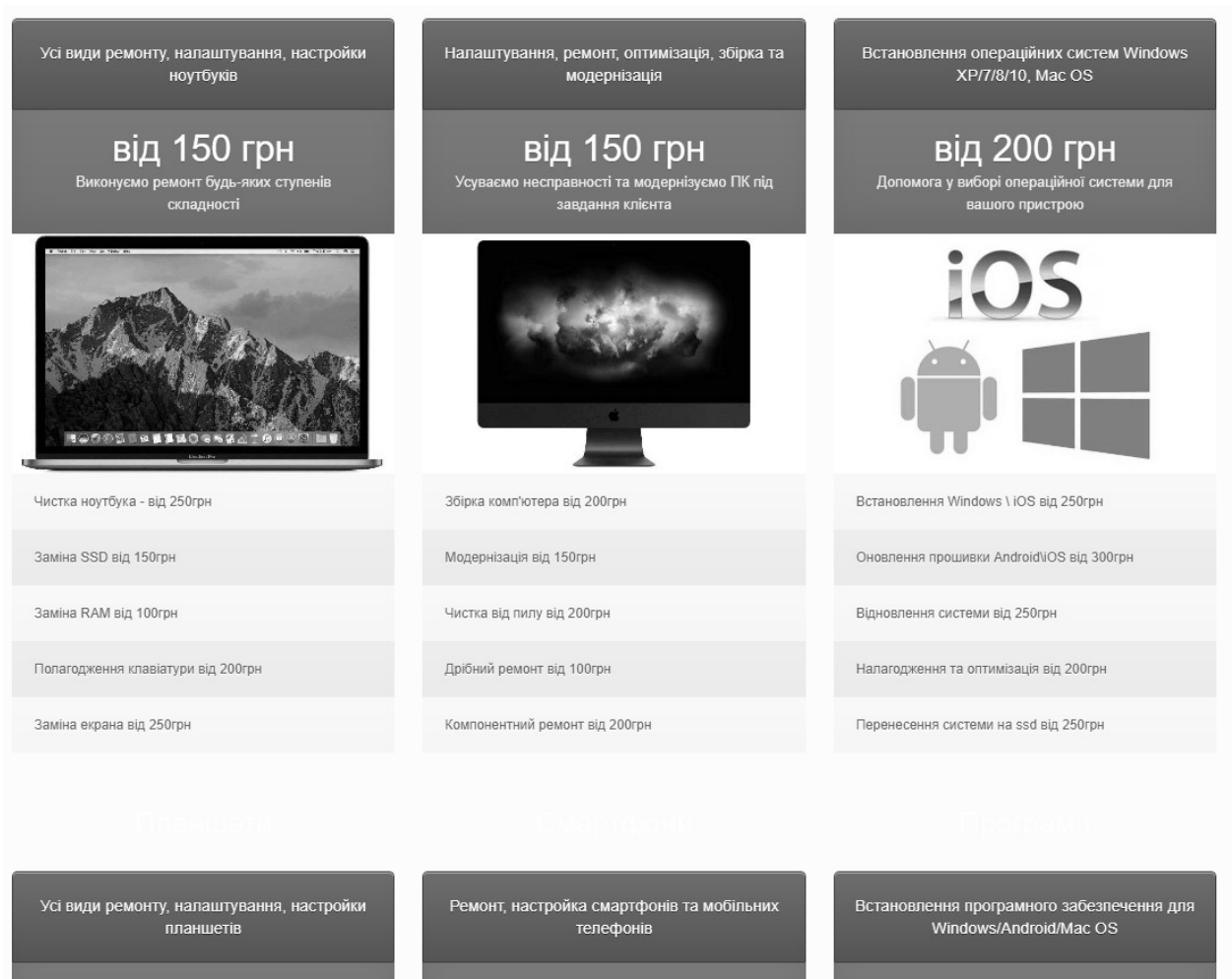


Рис. 3.3. Зовнішній вигляд блоку

Далі по плану треба зробити блок «Як ми працюємо» з детальним ланцюгом подій після потрапляння клієнта на сайт. Вона буде зроблена у вигляді п'яти іконок з написами подій, оформлена текстом зверху та такою ж decorate лінією, як раніше було оформлене головне вікно сайту:

```
<div class="sow-headline-container ">
<h4 class='sow-headline'>ЯК МИ ПРАЦЮЄМО</h4>
```

```

<div class="decoration">
<div class="decoration-inside"> </div> </div>
<h4 class='sow-sub-headline'>Ми працюємо заради клієнта.
Індивідуальність та хороша якість - це про нас.</h4>
</div>

```

Далі робимо іконки з написами:

```

<div class="sow-features-list sow-features-responsive">
<div class="sow-features-feature sow-icon-container-
position-right sow-features-feature-last-row" style="width:
33.333%">
<div class="sow-icon-container sow-container-rounded-square"
style="color: #dd9933; ">
<span class="sow-icon-elegantline" data-sow-icon="&#xe004;"
style="font-size: 57px; color: #FFFFFF"></span></div>
<div class="textlanzug">
<h5>Клієнт залишає заявку на сайті</h5></div></div></div>

```

Тепер сайт має першу частину ланцюгу роботи (рис. 3.4.).

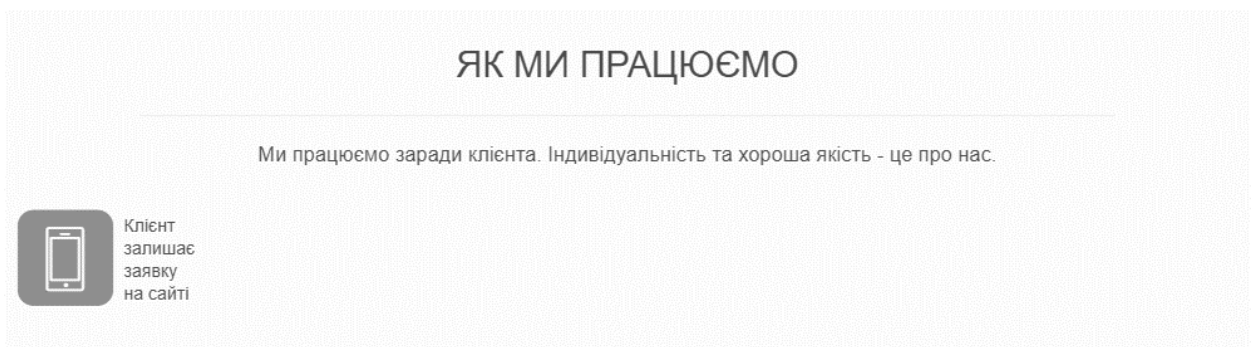


Рис. 3.8. Зовнішній вигляд незакінченого блоку

Робимо ще 4 блоки та розміщуємо їх справа, за допомогою CSS рівняємо (Рис. 3.5.).



Рис. 3.5. Зовнішній вигляд повного блоку

Наступним кроком буде вікно відгуків та пропозицій. В першу чергу напишемо анотацію блоку, як раніше і вставимо ту ж декорацію:

```
<div class="sow-headline-container ">
<h4 class='sow-headline'>ЩО ГОВОРЯТЬ ПРО НАС КЛІЄНТИ</h4>
<div class="decoration">
<div class="decoration-inside"> </div> </div>
<h4 class='sow-sub-headline'>Відгуки та пропозиції</h4>
</div>
```

Тепер потрібно зробити саме віконце відгуку. Воно буде мати вигляд темного прямокутника із ім'ям та текстом. При наведенні буде видно дві стрілки та три крапки. Це дасть змогу користувачеві переглянути відгуки з трьох доступних у кожному віконці:

```
<ul class="sow-slider-images" data-
settings="{&quot;pagination&quot;:true,&quot;speed&quot;:800,&qu
ot;timeout&quot;:8000,&quot;paused&quot;:false,&quot;pause_on_ho
ver&quot;:false,&quot;swipe&quot;:true,&quot;nav_always_show_mob
ile&quot;:&quot;&quot;,&quot;breakpoint&quot;:&quot;780px&quot;}
">
<li class="sow-slider-image sow-slider-image-cover"
style="background-image:
url(wp-content/uploads/2021/01/0tzyvy1.png)">
```

```

<div class="sow-slider-image-container">
<div class="sow-slider-image-wrapper">
<p>Владислав</p>
<p>Відгук</p>
</div></div></li></ul>

```

Таким чином зробимо 3 відгуки, після чого дамо їм змогу перегортатись:

```

<ol class="sow-slider-pagination">
<li><a href="#" data-goto="0" aria-label="display slide
1"></a></li>
<li><a href="#" data-goto="1" aria-label="display slide
2"></a></li>
<li><a href="#" data-goto="2" aria-label="display slide
3"></a></li>
</ol>

```

Тепер вони мають змогу перегортатись, але користувач ще не може їх перегортати. Для цього зробимо стрілки:

```

<div class="sow-slider-base " style="display: none">
<div class="sow-slide-nav sow-slide-nav-next">
<a href="#" data-goto="next" aria-label="next slide" data-
action="next">
<em class="sow-sld-icon-thin-right"></em>
</a>
</div>
<div class="sow-slide-nav sow-slide-nav-prev">
<a href="#" data-goto="previous" aria-label="previous slide"
data-action="prev">
<em class="sow-sld-icon-thin-left"></em>
</a>

```

```
</div></div>
```

Тепер користувач може передивитись усі слайди з відгуками, але HTML 5 не підтримує самостійне циклування слайдів, тому будемо використовувати для цього плагін jQuery Cycle2. Для його використання завантажимо файл з кодом та ініціалізуємо скрипт за допомогою цих команд:

```
<script type='text/javascript' src='wp-content/plugins/sow-
widgets-bundle/js/jquery.cycle.min.js?ver=1.17.11' id='sow-
slider-slider-cycle2-js'></script>
<script type='text/javascript' src='wp-content/plugins/sow-
widgets-bundle/js/slider/jquery.slider.min.js?ver=1.17.11'
id='sow-slider-slider-js'></script>
```

Після налаштування слайди будуть самостійно перегортуватись, а користувачу залишається лише читати:

У футері треба додати гугл мапу, контакти, соціальні мережі та знак копірайту. В першу чергу мапа. Гугл має функцію вбудовування мапи в HTML код, тож скористуємось нею. Для цього треба зайти до гугл мап та відзначити місце, яке має бути фізичним розташуванням сервісу. Поки що відзначимо випадкове місце на мапі та вбудуємо її:

```
<div class="sow-headline-container ">
<h4 class='sow-headline'>ЯК НАС ЗНАЙТИ</h4>
<div class="decoration">
<div class="decoration-inside"></div>
</div>
<h4 class='sow-sub-headline'>м. Кривий Ріг, Вулиця
Володимира Великого, 148/2 </h4>
</div></div>
```



```

<div align="center"><div class="map"><iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!
1d668.0978411828271!2d33.42899845307992!3d47.9481556394034!2m3!
1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x40db200b784cd22f
%3A0xb691b4d6897c179!2z0LLRg9C70LjRhtGPINCS0L7Qu9C-
0LTQuNC80LjRgNCwINCS0LXQu9C40LrQvtCz0L4sIDE00Cwg0JrRgNC40LLQuNC5
INCg0ZbQsywg0JTQvdGW0L_RgNC-
0L_QtdGC0YDQvtCy0YHRjNC60LAg0L7QsdC70LDRgdGC0YwsIDUwMDAw!5e0!
3m2!1sru!2sua!4v1622565347381!5m2!1sru!2sua" width="2800"
height="600" style="border:0;" allowfullscreen=""
loading="lazy"></iframe></div></div >

```

Тепер на сайті є мапа і у користувачів не виникне питання куди везти їх техніку (Рис. 3.6.).



Рис. 3.6. Гугл мапа на web-сайті

Залишилось створити футер за допомогою

```

<footer id="colophon" class="site-footer"
role="contentinfo">

```

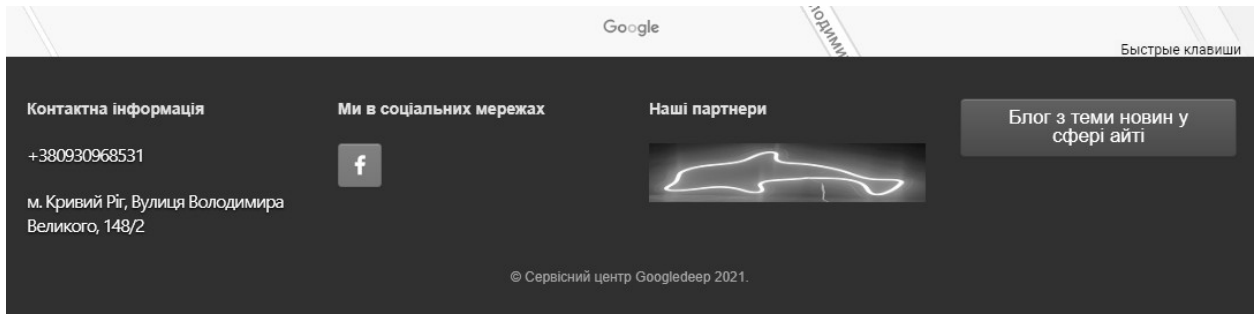



Рис. 3.7. Зовнішній вигляд футеру

Сторінку цін та послуг почнемо в стилі головної, добавимо лого та поміняємо колір заднього фону за допомогою команди:

```
<li class="sow-slider-image" style="background-color: #404040"></li>
```

Організуємо елементи web-сторінки за допомогою тега table, який допоможе створити таблицю із найменуваннями послуг та їх вартістю:

```
<h2 class="tablepress-table-name tablepress-table-name-id-1">Основні послуги</h2>
<table id="tablepress-1" class="tablepress tablepress-id-1">
  <thead>
    <tr class="row-1 odd">
      <th class="column-1">Вид робіт</th><th class="column-2">Вартість, грн.</th>
    </tr>
  </thead>
  <tbody class="row-hover">
    <tr class="row-2 even">
      <td class="column-1">Діагностика/усунення несправності в апаратній частині ПК</td><td class="column-2">від 100</td>
    </tr>
    <tr class="row-3 odd">
      <td class="column-1">Чистка системного блоку від пилу</td><td class="column-2">від 100</td>
    </tr>
    <tr class="row-4 even">
```

```
<td class="column-1">Встановлення та підключення ПК</td><td
class="column-2">Від 100</td>
</tr></tbody></table>
```

І так далі можливо створити будь-яку множину таблиць. По тій же системі створимо сторінку прайс-листу фото та заповнюємо їх таблицями на смак (рис. 3.8.).

В кінці сторінок робимо футер і мапу, як на головній. Над мапою зробимо дві кнопки – телефон та кнопку повернення до головної сторінки. Щоб зробити функціональну кнопку телефону для користувачів мобільних пристроїв скористаємося функцією:

```
<a href="tel:+380930968531" class="ow-icon-placement-left
ow-button-hover">
<span>
<span class="sow-icon-fontawesome sow-fas" data-sow-
icon="&#xf87b;"></span>
+380930968531</span></a>
```



Вид робіт	Вартість, грн.
Діагностика/усунення проблем в апаратній частині ПК	від 100
Чистка системного блоку	від 100
Установка та підключення ПК	від 100
Доставка клієнту частин	від 50
Консультація з питань ПК	Безкоштовно
Простий фахівець(тільки з вини замовника)	150
Навчання або консультування	Договірна
Налагодження або оптимізація Windows	від 100

Рис. 3.8. Зовнішній вигляд прайс-листу на роздруківку фото

За допомогою функції “href=” tel: “” можна зробити посилання на телефонний номер та одразу зробити дзвінок.

За допомогою мови HTML були створені основні сторінки web-сайту. Була виконана робота з оформлення всіх написів та формування красивого, але консервативного дизайну. Усі посилання були оформлені у відповідності з посібниками [9]. Web-сайт успішно функціонує, тепер його треба протестувати та уявити себе користувачем, який нічого не знає про структуру сайту.

3.2. Функціональне та юзабіліті-тестування

В процесі юзабіліті-тестування потрібно перевірити web-сайт на помилки при очікуваних діях користувача. При потраплянні на сайт користувач бачить головну сторінку з логотипом компанії та її спеціалізацією (див. рис. 3.2).

Першим чином треба перевірити всі кнопки головної сторінки на функціонал. Після перевірки можна упевнитись в тому, що кнопки працюють і переносять користувача до інших сторінок сайту, що відповідає карті сайту.

В якості перевірки масштабування web-сайту спробуємо зменшити сайт до 25% (рис. 3.9.) та збільшити його до 175% (рис. 3.10.), сайт повинен масштабуватись та розміщуватись по центру екрану, при цьому елементи сайту не повинні зникати.



Рис. 3.9. Масштабування 25%, сайт успішно масштабувався

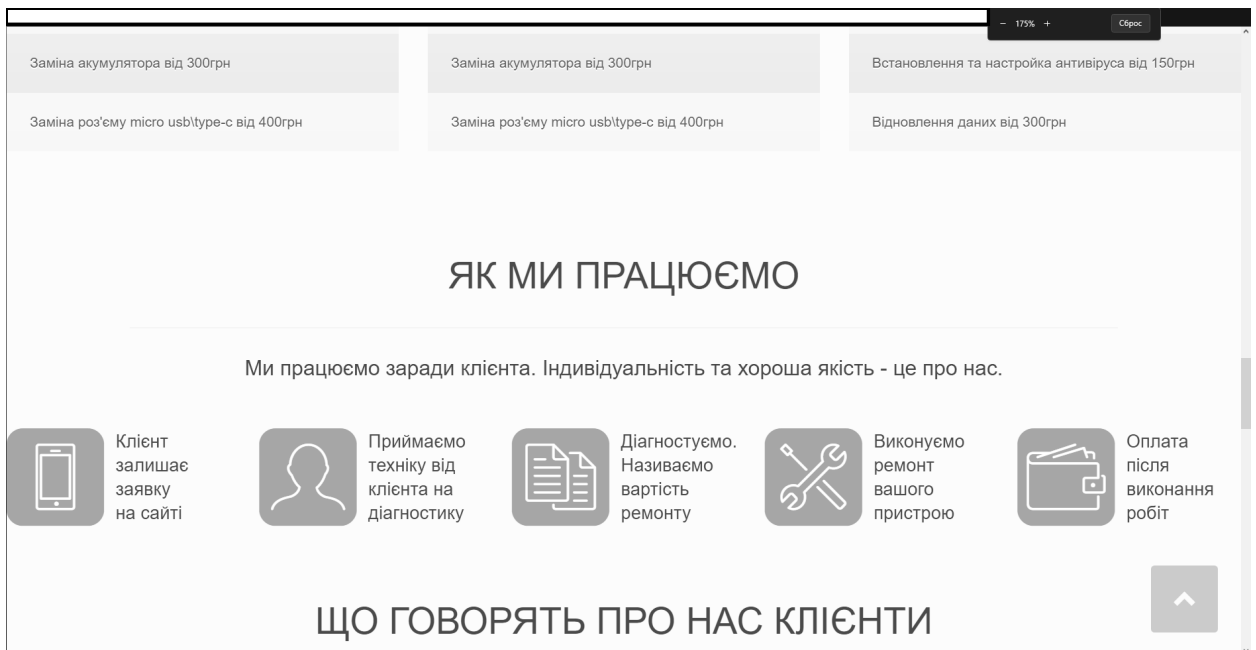


Рис. 3.10. Масштабування 175%, елементи сайту не зміщено

Сайт успішно пройшов тестування масштабування, далі перевіримо кросс-браузерність. Оскільки сайт розроблявся в Opera, можна вважати його перевіреним. Далі потрібно перевірити web-сайт в Google Chrome (рис. 3.11.).

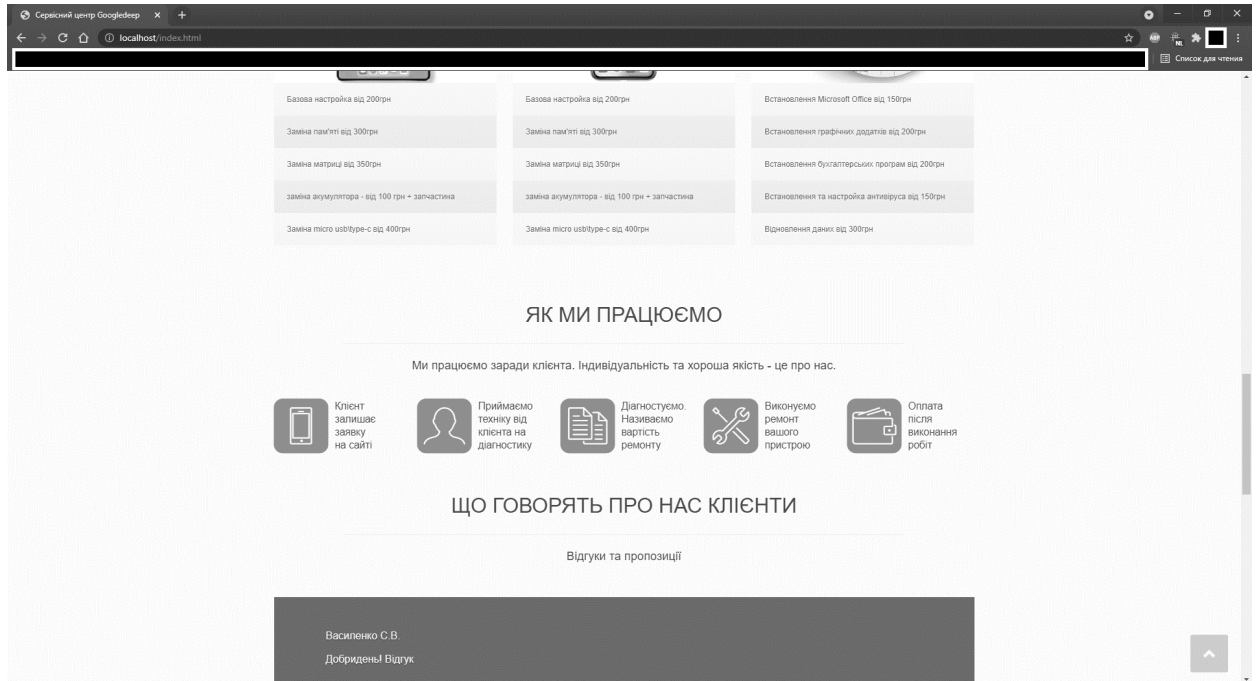


Рис. 3.11. Web-сайт в Google Chrome

Сайт правильно відображається в іншому браузері, були перевірені всі сторінки та система реєстрації. Продовжимо перевірку сайту перевіркою системи реєстрації. Для цього треба натиснути на кнопку «Увійти» та увести в форму реєстрації свої бажані дані. Для перевірки були введені стандартні дані «Admin» з уточнюванням поля (рис. 3.12.).

Форма реєстрації

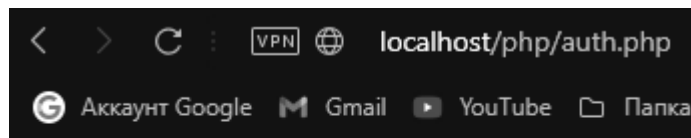
Рис. 3.12. Форма реєстрації з даними

Тепер переходимо до PhpMyAdmin та перевіряємо присутність даних у таблиці (Рис. 3.13.).

				id	login	pass	name			
<input type="checkbox"/>		Edit		Copy		Delete	9	Мій логін	Мій пароль	Василь Петрович
<input type="checkbox"/>		Edit		Copy		Delete	10	Мій логін	Мій логін	name
<input type="checkbox"/>		Edit		Copy		Delete	11	МійлОгін	МійлОгін	Ім'Я
<input type="checkbox"/>		Edit		Copy		Delete	12	ЛоГіН	ЛоГіН	ЛоГіН
<input type="checkbox"/>		Edit		Copy		Delete	13	Adminlogin	Adminpassword	Adminname

Рис. 3.13. SQL таблиця користувачів (users)

13 користувач успішно зареєстрований і дані введені правильно, тепер система реєстрації майже перевірена. Тепер видалимо цей запис та спробуємо авторизуватись за цими даними. В результаті цих дій отримуємо помилку (рис. 3.14.).



Цей користувач не є дійсним

Рис. 3.14. Помилка авторизації

При спробі створити реєстраційний запис та авторизуватись з неправильним реєстром символів сайт видасть ту ж помилку.

При спробі створити запис із логіном або паролем «123456789012345678901234567890123», тобто 33 символи сайт переходить на сторінку помилки «Неприпустима довжина логіна (від 2 до 30 символів)».

Тепер система реєстрації та авторизації перевірена, можна авторизуватись та перевірити систему повідомлення. Для цього після авторизації на головній сторінці виникає дві кнопки (Рис. 3.15.).

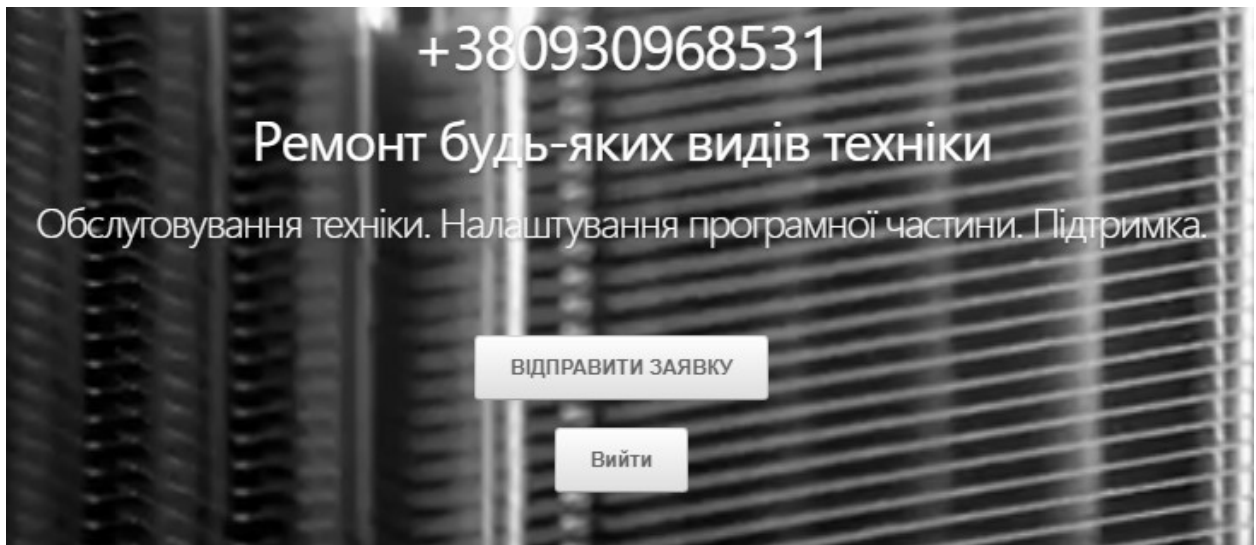


Рис. 3.15. Головна сторінка для авторизованого користувача

Користувач натискає на кнопку «Відправити заявку» та сайт переносить користувача на сторінку повідомлення. Надішлемо тестове повідомлення (Рис. 3.16.) та перевіримо його у таблиці:

Ваше повідомлення

+38012312312

VladPlahiy@gmail.com

Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю питання Я маю

Надіслати

Рис. 3.16. Тестове повідомлення

Переходимо до таблиці і можемо повністю прочитати повідомлення (Рис. 3.17.). Після цього система повідомлення протестована, а користувач автоматично повернувся до головної сторінки.

Для перевірки працездатності сайту були перевірені всі керовані елементи:

- 1) кнопки
- 2) поля введення
- 3) зв'язок з SQL таблицею
- 4) зовнішній вигляд web-сайту при різних діях користувача

Для перевірки кожного елемента бралось 2 набору даних - правильний і завідомо неправильний.

В результаті перевірки вірний набір даних повинен бути коректно оброблений і повинні бути отримані результати обробки. Невірний набір даних повинен бути відфільтрований і повинна бути видана відповідна помилка.

Результати тестування представлені у таблиці 3.1:

Таблиця 3.1

Таблиця результатів тестування web-сайту

Об'єкт тестування	Очікуваний результат	Отриманий результат відповідає очікуваному
Працездатність кнопок	Кнопки працюють, відповідають посиланням.	+
Масштабування	Сайт правильно масштабується, контент не зникає.	+
Кросс-браузерність	Сайт повинен правильно відображатись у різних браузерах.	+
Система реєстрації	Дані користувача вносяться до SQL таблиці, реєстр відповідає введеному, при видаленні користувача він повністю видалюється та не має можливості увійти.	+
Система авторизації	При введенні логіну або паролю більше 30 символів користувачу буде видана	+

	помилка. Система повинна коректно перевіряти присутність комбінації логіну. та паролю в SQL таблиці та виконувати відповідні дії	
Система повідомлення	У таблицю повинні заноситися символи, характерні для телефону та електронної адреси, таблиця повинна підтримувати повідомлення до 9999 символів.	+
Тестове повідомлення	Тестове повідомлення повинно коректно зберігатись до системи, адміністрація повинна мати змогу комфортно переглянути велике повідомлення.	+
Система виходу	Користувач повинен мати змогу вийти з системи, введене при реєстрації ім'я повинно відобразитись в повідомленні.	+

Система web-сайту пройшла функціональне та юзабіліті-тестування. Сайт готовий до роботи після зміни шаблонних картинок на дизайнерські логотипи.

3.3. Просування сайту

Щоб сайт мав хорошу позицію у пошукових системах, його треба просувати. Часто за просування сайту відповідає звичайна реклама, але це дорого і довго, тому існує SEO (аббревіатура від Search Engine Optimization) - комплекс заходів щодо збільшення видимості сайту в пошукових системах за цільовими пошуковими запитам.

Для висококонкурентних комерційних тематик дуже важлива SEO оптимізація. По статистиці на 2021 рік SEO генерує більш ніж 50% цільового трафіку для комерційних сайтів.

У загальному випадку, для отримання віддачі після запуску робіт по просуванню сайту потрібно від 3 тижнів до 9-12 місяців. Терміни визначаються рівнем конкуренції за запитом і загальному рівню конкуренції в просуваємій тематиці. Також істотно позначаються на строках такі фактори як:

- 1) вік сайту (критерій довіри до сайту з боку пошукової системи);
- 2) число сайтів, які активно просуваються в даній тематиці (критерій «щільності» результатів видачі);
- 3) загальний трафік на сайті (критерій оцінки корисності сайту для користувачів);
- 4) число і якість ресурсів, які посилаються на просувний сайт (критерій авторитетності ресурсу);
- 5) кількість релевантних запиту документів на сайті (оцінка відповідності тематики сайту пошуковому запиту).

Однак SEO-спеціалісти дорого коштують, тому що місце у рейтингу пошукової системи напряму впливає на кількість клієнтів. Психологія людини не хоче перегортати десятки сторінок пошукової системи аби потрапити на сайт, вона перевірить лише першу сторінку.

Чому SEO просування розвивається з кожним роком? Тому що Google знає, що правка сайту та хмарний хостинг, CDN і інші настройки дорого коштують. Як початковий сайт будемо використовувати «All in One SEO Pack». Для цього лише загрузимо його до системи та дамо змінити декілька посилань, створити ключові слова для невеликого просування сайту у пошуковій системі. Це робиться у самій програмі, вона міняє програмний код так, щоб у ньому було більше посилань на спеціалізацію сайту.

В результаті функціонального тестування сайту не було виявлено помилок. Сайт протестований та готовий до роботи. В результаті роботи з SEO оптимізацією у код сайту були внесені теги та ключові слова, але після

виходу до інтернету сайт потребує не один місяць для залучення великого числа клієнтів.

ВИСНОВКИ

Під час аналізу популярних мов програмування для розробки web-сайтів було проведено їх порівняння та вибрано найбільш актуальні засоби починаючих розробників. Вибір пріоритетних засобів розробки проходив за двома критеріями: доступність та функціональність.

При аналізі існуючих мов програмування було проведено їх порівняння та виділені їх переваги та недоліки.

Після вибору засобів розробки було розпочато вивчення PHP та розробка самого проекту. В ході розробки було придбано навички роботи у середовищі розробки Brackets, вивчено принципи роботи мови програмування PHP та мови розмітки HTML 5. Було вивчено команди та принципи роботи СУБД MySQL та принцип хостування сайту на домашньому сервері для роботи.

PHP відноситься до мов загального призначення. Проте ця мова перш за все застосовується для розробки web-додатків. Це серверна мова програмування, тобто її команди виконуються на сервері. Написані на PHP програми отримують дані від користувачів сайту, обробляють їх, взаємодіють з базами даних, повертають на сайт оброблену інформацію.

Спочатку аббревіатура PHP позначала Personal Home Page. Це відображає основне призначення мови - створення web-сторінок. На PHP написані популярні системи управління контентом (CMS), наприклад, WordPress, Joomla !, Drupal. На одному тільки WordPress працює близько третини від усіх сайтів в інтернеті. Це підтверджує актуальність і популярність PHP. Цією мовою написані і популярні фреймворки для створення сайтів, наприклад, Laravel, Yii2, Symfony. Нарешті, такі масштабні і популярні проекти, як Facebook і «ВКонтакте», теж створені на PHP.

Проміжний підсумок: PHP - інтерпретована мова з динамічною типізацією. Застосовується переважно для розробки web-додатків. За

результатами травня 2022 року PHP входить в десятку найпопулярніших мов програмування в індексі ТЮВЕ. Ця мова посідає дев'яте місце, випереджаючи Swift, Ruby, Objective-C, Go. [10].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет-технології в бізнесі: навч. посіб. / О. І. Щедрина, М. М. Агутін. – К. КНЕУ, 2012. – 303 с. ISBN 978-966-483-584-5
2. Роберт Шелдон, Джофрей Моей MySQL 5: базовий курс: Пер. с англ. – Киев: Диалектика, 2007. – 880 с.
3. [Дино Эспозито](#) Разработка современных веб-приложений: анализ предметных областей и технологий. - М.: Вильямс, 2017.- 464 с.
4. Методичні вказівки до самостійного вивчення системи управління базами даних MySQL. Частина 2. Команди модифікації та доступу до даних мови СУБД MySQL. – Кривий Ріг: KEI ДВНЗ «КНЕУ ім. В.Гетьмана», 2008.
5. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов /Д. Дакетт.: Пер. с англ. – Киев: Диалектика, 2019. – 480 с.
6. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5. 4-е вид. - СПб.: Пітер, 2016. - 768 с : іл. - (Серія «Бестселери O'Reilly »). ISBN 978-5-496-02146-3
7. Лоусон Б., Шарп Р. Вивчаємо HTML 5. Бібліотека спеціаліста. - СПб.: Пітер, 2011. – 272 с.: іл. – (Серія «Бібліотека спеціаліста»). ISBN 978-5-459-00269-0
8. Дронов В. А. HTML 5, CSS 3 та Web 2.0. Розробка сучасних Web-сайтів. СПб .: БХВ-Петербург, 2011. -416с .: іл .- («Професійне програмування») ISBN 978-5-9775-0596-3
9. TIOBE [Електронний ресурс]: © 2021 TIOBE Software BV Режим доступу: <https://www.tiobe.com/tiobe-index/>.
- 10.Htmlbook [Електронний ресурс] © 2002–2021 Htmlbook Режим доступу: <http://htmlbook.ru/>.
- 11.Алексеев А.П. Введение в Web-дизайн: учебное пособие. - М.: СОЛОН-ПРЕСС, 2008.

12. Ульман Л. MySQL. Руководство по изучению языка: Пер. с англ. – Москва, Санкт-Петербург: ДМК Пресс, Питер, 2004. 352 с.
13. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript: Пер. с англ. – Санкт-Петербург: Питер, 2011. – 496 с.
14. Ричард Вагнер, Аллен Вайк JavaScript. Энциклопедия пользователя: Пер. с англ. – Киев: ДИАСофт, 2001. – 464 с.
15. Хольцнер С. PHP в примерах. – М.: Высш. шк., 2014. – 473 с.
16. Майерс Г., Баджетт Т. Искусство тестирования программ. Пер. с англ. – М.: Издательский дом «Вильямс», 2012. – 272 с.
17. Беллиньясо М. Разработка Web-приложений: Задача-проект-решение. М.: Изд. дом «Вильямс», 2007. – 640 с.
18. Корпоративные порталы. Анализ, выбор продукта, практика внедрений. – Enterprise Portals: Все о корпоративных порталах. – [Электронный ресурс]. – URL: <http://www.enterprise-portals.ru/index.php>.
19. Шишкина Л. Корпоративный портал – оптимальное решение для организации рабочего места сотрудников. – Обучающееся Сообщество менеджеров. – [Электронный ресурс]. – URL: <http://www.e-ccutive.ru/community/articles/1372883/>
20. Створення сайтів. – Студія веб-дизайну. – [Электронный ресурс]. – URL: <http://www.wdesign.net.ua/ua/#>.
21. Порталы. – КСК разработчик заказного ПО. – [Электронный ресурс]. – URL: <http://www.kck.ru/kcksite/kcksite.nsf/pages/portals.html>.
22. Алгоритм – Создание сайта. – О том, как бесплатно создать свой сайт. – [Электронный ресурс]. – URL: <http://webhelp.ucoz.ru/publ/2>.

ДОДАТКИ