

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«_____» _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 123 - “Комп'ютерна інженерія”

Науковий керівник роботи:

_____ (підпис)

Щербаков Є.В.

(ініціали, прізвище)

Консультант з охорони праці:

_____ (підпис)

Критська Я.О.

(ініціали, прізвище)

Студент:

_____ (підпис)

Стиров С.Е.

(ініціали, прізвище)

Група:

КІ-17ДМ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень магістр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 123 - "Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
І.С. Скарга-Бандурова
« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Стиров Сергій Едуардович
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів.

керівник проекту (роботи) Щербаков Євген Васильович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від " " _____ 2019 р. № _____
2. Строк подання студентом роботи 07.01.2019
3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Визначення сучасних принципів побудови інформаційно-управляючої системи; Аналіз сучасних програмних-технічних засобів для побудови комп'ютерної системи обліку і продажу; Аналіз сучасних методів оптимізації; Практична реалізація системи обліку і продажу транспортних засобів; Охорона праці;

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Я.О.		

7. Дата видачі завдання

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Пошук літературних джерел	06.09.2018 – 20.09.2018	
2.	Аналіз актуальних методів розробки інформаційно-управляючої системи	21.09.2018 – 12.10.2018	
3.	Аналіз актуальних методів пошукової оптимізації	15.10.2018 – 30.10.2018	
4.	Практична реалізація системи обліку і продажу	01.11.2018 – 10.12.2018	
5.	Розробка заходів з охорони праці	11.12.2018 – 17.12.2018	
6.	Оформлення пояснювальної записки	23.12.2018 – 06.01.2019	

Студент

_____ (підпис)

Стиров С.Е.

_____ (прізвище та ініціали)

Науковий керівник

_____ Щербаков Є.В

АНОТАЦІЯ

Стиров С. Е. Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів.

Проаналізовані сучасні вимоги до створення інформаційно-управляючих систем. Досліджені сучасні методи та стратегії створення систем обліку і продажу. Практично реалізована система обліку і продажу транспортних засобів.

Ключові слова: інтернет, система, інформаційно-управляюча, сервіс, веб-сторінка, розмітка, інтерфейс, база даних, html, css, js, php, mvc.

АННОТАЦИЯ

Стыров С.Э. Исследование и разработка компьютерной системы учета и продажи транспортных средств.

Проанализированы современные требования к созданию информационно-управляющих систем. Исследованы современные методы и стратегии создания систем учета и продажи. Практически реализована система учета и продажи транспортных средств.

Ключевые слова: интернет, система, информационно-управляющая, сервис, веб-страница, разметка, интерфейс, база данных, html, css, js, php, mvc.

ABSTRACT

Styrov S.E. Research and development of a computer system for accounting and selling vehicles.

Analyzed the modern requirements for the development of information management systems. Modern methods and strategies for creating accounting and sales systems have been studied. Practically implemented a system of accounting and sales of vehicles.

Keywords: Internet, system, information management, service, web page, markup, interface, database, html, css, js, php, mvc.

Зміст

ВСТУП	7
РОЗДІЛ 1 СУЧАСНИЙ СТАН МЕТОДІВ ПОБУДОВИ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ	10
1.1 Вимоги до побудови інформаційно-управляючої системи	10
1.2 Структура та особливості побудови інформаційно-управляючої системи в мережі інтернет	11
1.3 Способи створення інформаційно-управляючих систем.....	13
1.4 Вимоги до пошукової оптимізації	13
1.5 Поняття електронної комерції та її основні складові	17
1.6 Аналіз сучасних робіт	17
1.7 Формулювання проблеми дослідження.....	19
1.8 Висновки до розділу 1	19
РОЗДІЛ 2 АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ ОБЛІКУ І ПРОДАЖУ	21
2.1 Аналіз типів й основних характеристик серверів.....	21
2.1.1 Основні типи серверів.....	21
2.1.2 Актуальні серверні рішення.....	23
2.2 Клієнт-серверна взаємодія і ролі серверів	24
2.3 Порівняння веб-серверів Apache та Nginx	25
2.4 Аналіз актуальних методів розробки інформаційно-управляючої системи.....	26
2.5 Аналіз технологій розробки клієнтської частини системи.....	27
2.5.1 Адаптивна верстка	28
2.5.1.1 Технологія адаптивної сітки Bootstrap.....	31
2.5.2 Мова гіпертекстової розмітки HTML5	31
2.5.3 Каскадні таблиці стилів CSS3.....	33
2.5.3.1 Препроцесор SASS.....	34
2.5.4 Мова програмування JavaScript.....	35
2.6 Аналіз технологій розробки динамічних систем.....	35

	5
2.6.1 Мова серверних сценаріїв PHP	37
2.6.2 Патерн Model-View-Controller	38
2.6.3 Система управління контентом	39
2.6.3.1 Порівняння популярних систем управління контентом	40
2.7 Аналіз положень індексування веб-системи.....	41
2.8 Аналіз роботи популярних пошукових систем	43
2.9 Висновки до розділу 2.....	45
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ З ОБЛІКУ І ПРОДАЖУ ТРАНСПОРТНИХ ЗАСОБІВ	47
3.1 Проектування архітектури системи обліку і продажу	47
3.2 Розробка дизайну користувальницької частини системи.....	48
3.3 Розробка та реалізація головної сторінки	51
3.4 Розробка серверної частини системи.....	52
3.5 Розробка системи реєстрації та входу	54
3.6 Реалізації сторінки продажу транспорту.....	56
3.7 Реалізація сторінки управління системою	59
3.8 Висновки до розділу 3.....	64
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ	65
4.1 Загальні питання з охорони праці	65
4.2 Аналіз стану умов праці.....	65
4.2.1 Вимоги до приміщень.....	66
4.2.2 Вимоги до організації місця праці	66
4.2.3 Навантаження та напруженість процесу праці	67
4.3 Виробнича санітарія	68
4.3.1 Аналіз небезпечних та шкідливих факторів при експлуатації.....	68
4.3.2 Пожежна безпека	70
4.3.3 Електробезпека.....	70
4.4 Гігієнічні вимоги до параметрів виробничого середовища	70

4.4.1 Мікроклімат.....	70
4.4.2 Освітлення	71
4.4.3 Вентилювання	73
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	73
4.6 Охорона навколишнього природного середовища	76
4.6.1 Загальні дані з охорони навколишнього природного середовища	76
4.6.2 Визначення впливу та заходів щодо поводження з відходами ІТ галузі.....	76
4.7 Висновки до розділу 4.....	76
ВИСНОВКИ.....	78
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	81

ВСТУП

Актуальність теми. Ще двадцять років тому "Всесвітня павутина", якій в 2018 році виповнилось 49 років, використовувалася практично тільки для пошуку необхідної інформації[1]. З кожним днем, число користувачів Інтернету, нових ресурсів і можливостей використання мережі стрімко зростає. На сьогоднішній день кількість інтернет-ресурсів перевищує один мільярд, а число користувачів за підрахунками на 2017 рік становила більше трьох мільярдів осіб, близько 45% населення землі.

Можливості інтернету ростуть швидше, ніж кількість сайтів і користувачів. Вже нікого не дивує, що через мережу можна заробляти гроші, сплатити комунальні послуги, замовити їжу онлайн, купити одяг або навіть автомобіль. Глобальна мережа сьогодні - це впливова сила: інструмент просунення, засіб представлення та пошуку інформації. І це всього 1% можливостей, які дає нам Всесвітня мережа.

Повертаючись до основної функції інтернету як джерела пошуку інформації, слід сказати, що це зумовило факт отримання даних про що завгодно і кого завгодно саме з мережі, не виходячи з власної квартири. Сьогодні кожен другий має обліковий запис як мінімум в одній соціальній мережі. Вся інформація про ту чи іншу людину, за рідкісним винятком, знаходиться у відкритому доступі. Кожен може без витрат і особливих умінь створити собі певний імідж за допомогою інтернету. Зазвичай, кожна компанія, фірма, організація використовує власні та сторонні веб-ресурси як спосіб представлення інформації і як засіб просування своїх послуг і товарів.

Ще кілька років тому вимоги до розробки веб-ресурсів були досить низькими. У їх оформленні не було анімації, зображень, різних динамічних елементів, а структура ресурсів була досить однотипною. Але з розвитком технологій "вага" і обсяг сайтів значно зросли. Зараз практично кожна друга компанія має власний ресурс в мережі. При цьому його структура є досить розгалуженою і на ньому може налічуватися кілька десятків або навіть сотень сторінок.

Розробка інтернет-ресурсу-тривалий і трудомісткий процес, який не можна проводити без ретельної підготовки. Перш ніж створювати сайт, слід продумати кожен його сторінку і елемент. Це необхідно, щоб забезпечити максимальну зручність для відвідувачів. Оскільки користувачі будуть потрапляти на різні сторінки з пошукових систем, слід надати їм можливість переходу між ними і вільного доступу до всієї інформації.

Користувач будує свою модель взаємодії з Інтернет-ресурсом на підставі вже наявного у нього досвіду. Саме це часто є вирішальним фактором при проектуванні структури сайту. У майбутньому популярність ресурсу буде залежати від того, наскільки очікування користувача узгоджуються з контентом, посиланнями і елементами, розміщеними на сторінці. Логічно

продумана і передбачувана структура сайту дозволить максимально спростити пошук необхідної інформації і зробить більш комфортною його експлуатацію.

Щоб користувач зайшов на потрібний сайт, використовується SEO - оптимізація (Search Engine Optimization) [2]. Цей інструмент являє собою сукупність заходів для підняття позиції сайту в результатах запитів пошукових систем. SEO, нарівні з деякими іншими технологіями інтернет-маркетингу, як, наприклад Social Media Marketing - просування в соціальних мережах і Search Engine Advertising - пошукова реклама, часто застосовуються разом в системі по просуванню ресурсу організації, компанії і т. д.

Статистика показує, що користувач навряд чи продовжить переглядати більше тридцяти позицій результатів запиту. Отже, сайту організації в принципі необхідно бути на перших місцях в пошукових системах, щоб про компанії дізналися і, тим більше, зацікавились предметом її діяльності. Тут починає працювати стереотип: чим вище позиція сайту, тим найбільш шанованим є цей веб-ресурс.

З точки зору теорії SEO-оптимізація, як елемент інтернет-маркетингу, є сферою достатньо вивченою, але часто обновлюваною. Дані навіть за 2017 рік в 2018 році можна вважати застарілими, що пов'язано з постійною модернізацією Інтернету [3]. Тим не менш, класичною залишається структура технології: існують мовний та технічний аспекти SEO-оптимізації. Перший відноситься до підбору ключових слів у запиті, самого контенту веб-ресурсу, посилення на інші портали і т. д. Розгляд пошукової оптимізації в даному ключі показує складність легкого на перший погляд способу просування сайтів, адже власний контент повинен бути унікальним, ключові запити – короткими і ясними і т. д.

Технічна складова SEO-оптимізації – це безпосереднє втілення мовного аспекту: розробка HTML-коду з урахуванням ключових слів і фраз, прописування тегів і т. д.

Сайт є одним із компонентів складної системи маркетингових заходів тієї чи іншої організації з просування свого бренду і предмету діяльності. Будучи досить ефективним засобом просування, в сукупності з використанням правильної структури та технології пошукової оптимізації, сайт являє собою сильний компонент маркетингового напрямку діяльності компанії. Таким чином, тема дипломної роботи “Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів” є актуальною на сьогоднішній день.

Мета і завдання дослідження. В цьому дипломному проекті досліджуються особливості побудови комп'ютерної системи інтернет-магазину з продажу транспортних засобів. Треба розглянути та проаналізувати структури веб-сайтів подібного призначення, методи пошукової оптимізації, основні технології, які використовуються при розробці подібних web-ресурсів й розробити систему обліку і продажу транспортних засобів.

Для досягнення мети необхідно виконати наступні завдання:

- проаналізувати методи й структуру побудови інформаційно-управляючих систем;
- описати цілі, що стоять перед інформаційно-управляючими системами;
- описати цілі, функції, види SEO-оптимізації;
- описати різновид серверів, основні характеристики серверів;
- проаналізувати сучасні зміни в алгоритмах пошукової оптимізації;
- визначити характеристики мовного і технічного аспектів розробки комп'ютерної системи інтернет-магазину;
- практично реалізувати комп'ютерну систему.

Об'єкт дослідження – інформаційно-управляючі системи.

Предмет дослідження – особливості побудови комп'ютерних систем інтернет-магазинів.

Методи дослідження. Для вирішення поставлених завдань було застосовано метод системного аналізу.

Практичне значення отриманих результатів. В ході роботи була проаналізована і створена комп'ютерна система обліку і продажу транспортних засобів. При розробці системи застосовані сучасні методи, що були проаналізовані в рамках даного дослідження.

Публікації. По даній роботі було зроблено дві публікації у збірнику наукових праць:

- Конференція: «Майбутній науковець», м. Сєвєродонецьк. Назва статті «Аналіз та особливості побудови комп'ютерної системи інтернет-магазину з продажу транспортних засобів» [28].

- Збірник: «ІТ-ідея», Назва статті «Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів » [29].

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконувалася протягом 2017-2018 рр. згідно з планами науково-дослідних робіт кафедри комп'ютерних наук та інженерії в межах НДР.

Структура і обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатку. Робота викладена на 99 сторінках машинописного тексту, містить 83 сторінок основного тексту, 37 рисунків, 7 таблиць, 2 додатки на 15 сторінках. Бібліографічний список включає 49 найменувань.

РОЗДІЛ 1

СУЧАСНИЙ СТАН МЕТОДІВ ПОБУДОВИ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ

1.1 Вимоги до побудови інформаційно-управляючої системи

Розробка інформаційно-управляючих систем – одна з найбільш затребуваних послуг в області інтернет-технологій. Послуги формування інформаційно-управляючих систем в цілому замовляють фірми, задіяні в різних областях бізнесу, індивідуальні підприємці [38].

Цілі, що стоять перед інформаційно-управляючими системами для організації:

1. Взаємодія з потенціальним споживачем.
2. Залучення нових покупців і партнерів.
3. Дані фірм час від часу змінюються, про новітні модифікації необхідно сповіщати покупців і нерідко це перетворюється в тягар, в разі якщо не застосовувати інформаційно-управляючі системи. На веб-сайті комфортно показувати всі анонси компанії, зміни вартості, послуг, обставин.
4. Організація товарів. Система як великий каталог з товаром, який можливо завжди доповнити, оновити.
5. Мінімальні затрати. Не потрібно мати великі будівлі з офісами. Потрібен тільки склад та невеликий штат співробітників.

Щоб система була успішною, вона повинна бути приваблива для відвідувачів. Тому в процесі розробки акцентується увага на таких особливостях:

- дизайн – це перший параметр, який оцінюють відвідувачі, тому він є одним з основних елементів;
- структура - повинна бути зручною, щоб кожен користувач зміг швидко знайти потрібну йому інформацію, гарний але незручний сайт буде приносити менше користі;
- контент – при відсутності цінної інформації ресурс не буде цікавий, але інформація повинна бути унікальною, цінною для користувачів і складатися з урахуванням вимог SEO;
- грамотність – структуровані тексти, інформативні підзаголовки, відсутність орфографічних, граматичних і стилістичних помилок не тільки додають проекту серйозність в очах користувачів, але і сильно впливають на просування сайту, так як враховуються пошуковими системами;
- швидкість завантаження - сторінки повинні відкриватися швидко, так як не всі користувачі готові миритися з повільним завантаженням сторінок, крім того, це негативно

впливає на просування, швидкість залежить від професійності на етапі верстки та програмування інформаційно-управляючої системи;

- просування – дозволяє збільшити кількість відвідувачів шляхом реєстрації сайту в каталогах, розміщення контенту, посилання і багато чого іншого, врахувати всі нюанси цього процесу може тільки професіонал в області SEO.

1.2 Структура та особливості побудови інформаційно-управляючої системи в мережі інтернет

Структура інформаційно-управляючої системи — це перелік розділів, сторінок, підрозділів і інших функціональних блоків, які можуть використовуватися в якості окремих елементів, а також їх взаємозв'язків. На стадії проектування немає необхідності піклуватися про зовнішній вигляд сторінок, оскільки картинки, текст і всі інші елементи дизайну будуть додані в майбутньому.

Особливості проектування структури інформаційно-управляючої системи:

- Необхідно дотримуватися одноманітність елементів.
- Слід ухилятися створення схожих сторінок. Якщо однотипна інформація може бути поміщена на одній сторінці, не слід розробляти для неї окрему.
- Не створювати розділи які дублюють один одного. Якщо функціонал будь-якого блоку частково повторюється в другому, потрібно позбавитися його. Не потрібно зайвий раз плутати відвідувачів ресурсу.
- Простий і швидкий доступ до всіх найбільш важливих розділів з усіх сторінок.
- Проектування інформаційно-управляючої системи повинно включати складання карти основних сторінок.

Структура інформаційно-управляючої системи поділяється на внутрішню і зовнішню. При цьому внутрішня структура часто значно впливає на зовнішню.

1. Внутрішня структура інформаційно-управляючої системи. До неї відносяться логічні зв'язки між різними сторінками ресурсу. В цій частині необхідно продумати, як користувач зможе максимально швидко отримати доступ до потрібної інформації. Наприклад, подбає щоб людині знадобилося не більше 3 натискань для переходу до всіх важливих розділів або цікавих сторінок. Також іноді внутрішньою структурою називають особливості розміщення директорій і ресурсів на сервері.

2. Зовнішня структура інформаційно-управляючої системи. Вона повністю повторює навігацію ресурсу і використовується для того, щоб спростити перехід відвідувачів по сторінкам. Завдяки їй людина може отримати доступ до основного функціоналу сайту з будь-

якої сторінки. При цьому зовнішня структура аналізується пошуковими системами і може вплинути на позицію системи у видачі.

Розрізняють чотири основні структури інформаційно-управляючої системи. У чистому вигляді кожна з них використовується досить рідко. Щоб зробити сайт максимально зручним для користувача і не нашкодити його функціоналу слід грамотно комбінувати різні види структур.

Основні структури:

1. Лінійна структура інформаційно-управляючої системи. Даний вид є найбільш простим. Подібна структура послідовна, в ній кожна зі сторінок веде на попередню і наступну сторінки ресурсу. В даному випадку навігація дуже проста і здійснюється з допомогою 2-3 посилань, використання цієї структури чимось схоже на перегортання сторінок книги;

2. Решітчаста структура. Розробка структури інформаційно-управляючої системи подібного типу також не відрізняється особливою складністю, але вона більш зручна для використання, ніж лінійний варіант. В даному випадку кожна сторінка ресурсу пов'язана з двома або трьома сторінками одночасно. Зазвичай кожна з них має зв'язки з однією або двома сторінками розділів одного рівня, а також з однією сторінкою — підрозділом. Тут вже повинна простежуватися ієрархічна структура інформації на ресурсі;

3. Ієрархічна структура інформаційно-управляючої системи. Основним елементом ієрархічної структури є головна сторінка. Посилання з неї ведуть на розділи другого рівня, а на сторінках другого рівня розміщені посилання на розділи третього рівня і т. д. В даному випадку користувач, перейшовши на головну сторінку, повинен обов'язково відвідати сторінку певного розділу, щоб дістатися до підрозділу. Найчастіше подібна структура застосовується для каталогів товарів;

4. Павутино-подібна структура. Вона передбачає зв'язок сторінок "всі з усіма" тобто користувач може потрапити з будь-якої сторінки ресурсу на будь-яку іншу сторінку, минаючи всі інші розділи. Розробка структури інформаційно-управляючої системи подібного типу досить складна, особливо якщо на ресурсі розташовано багато сторінок. На практиці користувач просто "загубиться" у величезній кількості посилань. Варто відзначити, що дана структура повторює в малому масштабі структуру всесвітньої мережі Інтернет.

Головна вимога до розробки структури — це логічність і простота. Користувач повинен з легкістю знаходити потрібні йому матеріали. Необхідно відзначити, що будь-який web-сайт не може постійно підтримувати одну і ту ж структуру. Не дивлячись на те, що розділи і каталоги будуть залишатися незмінними, при розміщенні нових матеріалів і статей буде виникати внутрішня перелінковка, яка вносить свої корективи.

1.3 Способи створення інформаційно-управляючих систем

Веб-програмування - це розділ програмування, орієнтований на розробку та створення веб-додатків, які забезпечують функціональність динамічних веб-сайтів. Програмування веб-додатків поділяється на два основних види: серверні і клієнтські. Серверні програми виконуються на стороні сервера, тобто того комп'ютера, на якому розміщена інформаційно-керуюча система. Вони виконуються ще до завантаження сторінок системи на комп'ютер користувача. У свою чергу, клієнтські програми виконуються на комп'ютері клієнта вже після завантаження сторінки з сервера і не вимагають її додаткової перезавантаження.

На даний час вже існує декілька способів розробки веб-додатків:

1. Створення системи за допомогою мов програмування. Цей спосіб створення передбачає самостійне написання коду, без використання готових шаблонів і набору функцій. Для цього в першу чергу потрібні знання мови гіпертекстової розмітки HTML, а також каскадних таблиць стилів CSS. Якщо ж планується динамічна система, то потрібно використовувати мову програмування PHP. Це найбільш затребувана технологія створення системи.

2. Конструктори. Принцип роботи з ними полягає у виборі шаблону, запропонованого конструктором, і підстроювання його під свої потреби. Процес створення і налаштування автоматизовані і оформлені у вигляді спеціальних програм-майстрів. Крім того, додавання нових і видалення існуючих елементів інтерфейсу максимально спрощено. Вони розташовуються на спеціальних панелях інструментів, доступних на екранах редагування інтерфейсу.

3. Фреймворки. У порівнянні з написанням систем з використанням мов програмування, даний спосіб створення додатків має перевагу у вигляді набору готових функцій, елементів оформлення і шаблонів.

4. Використання CMS. CMS - інструмент для тих, хто хоче створити сайт своїми руками і не володіє при цьому великими знаннями веб-програмування.

1.4 Вимоги до пошукової оптимізації

Існує багато визначень SEO, і не можна однозначно сказати, що те чи інше є вірним, але можна дати визначення SEO ось так - це інструмент просування інформаційно-управляючої системи, принцип роботи інструменту полягає в підвищенні місця виведення запитів пошуковими системами.

Під пошуковою оптимізацією розуміють комплекс заходів, спрямованих на підвищення позиції інформаційно-управляючої системи в пошуковій видачі по певних пошукових запитах. При цьому відповідність змісту сайту запитом називають релевантністю відповідності знайденого документа або набору документів інформаційним потребам користувача. Пошукова оптимізація включає в себе як покращення внутрішнього змісту, якості та кількості інформації, вдосконалення внутрішніх технічних аспектів реалізації інформаційно-управляючої системи, покращення візуальної привабливості для потенційних користувачів, так і роботу над зовнішніми чинниками.

Відповідно з дослідженням в 2018 році відсотки відвідувачів сайтів розподіляється наступним чином (рис.1.1):

1. Через пошукові системи – 76%.
2. По посиланню з іншого сайту - 8%.
3. Переходи з соціальних мереж – 7%.
4. Прямі заходи – 9%.

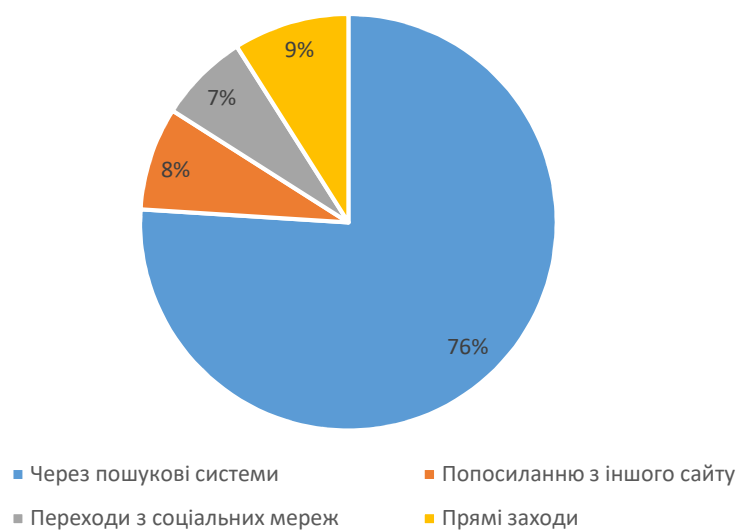


Рисунок 1.1 - Відсотки відвідувачів сайтів в 2018 році.

Метою SEO-оптимізації є, правильне пошукове просування інформаційно-управляючої системи і підвищення позиції в рейтингу пошукових систем, а також збільшення показника конверсії користувачів у клієнтів.

Процес оптимізації інформаційно-управляючої системи для пошукових систем полягає в певній роботі над контентом та його структурою, а також над внутрішніми та зовнішніми чинниками, що дозволяють пошуковим машинам, ґрунтуючись на поточній версії їх алгоритму, присвоювати сайту деяке місце в пошуковій видачі серед інших відомих йому сторінок. При

цьому фахівець, що займається оптимізацією, повинен орієнтуватися не тільки в поточній версії пошукового алгоритму тієї чи іншої пошукової системи, але і знати деякі психологічні аспекти поведінки користувачів в мережі Інтернет.

Досягнення поставленої мети зумовлює виконання таких завдань пошукової оптимізації, як:

1. Виявити потреби користувачів.
2. Скоригувати структуру інформаційно-управляючої системи.
3. Провести технічний аудит і виправити помилки.
4. Поліпшити уявлення інформаційно-управляючої системи в пошуку.

Слід також зазначити, що пошукову оптимізацію також іноді називають «просуванням», маючи на увазі просування певних сторінок як можна ближче до першого місця в пошуковій видачі по певному запиту. В цілому, пошукова оптимізація ділиться на два види. зовнішню і внутрішню оптимізацію (рис.1.2).

Внутрішня пошукова оптимізація - це набір дій і методів, спрямованих на поліпшення внутрішнього змісту і структури інформаційно-управляючої системи. До внутрішньої оптимізації безпосередньо відноситься:

- створення семантичного ядра, тобто підбор ключових слів, за якими інформаційно-управляюча система буде відображатися в результатах пошуку;
- редагування текстів, виправлення орфографічних або логічних помилок;
- технічна робота над структурою, створення карти інформаційно-управляючої системи, присвоєння сторінок легко запам'ятовуються і зрозумілих адрес, завдання інструкцій для пошукових роботів та інші методи, покликані спростити роботу як користувачам, так і пошуковим машинам;
- робота над дизайном і підвищення візуальної привабливості ресурсу;
- усунення технічних помилок, що уповільнюють роботу інформаційно-управляючої системи: починаючи від неіснуючих посилань і дублів сторінок, закінчуючи грубими семантичними і логічними помилками в коді сторінок, діючих на них скриптів, мережевих додатків.

До зовнішньої оптимізації належать такі методи, як нарощування посилальної маси на сайт і аналіз роботи, виробленої іншими оптимізаторами. Нарощування посилальної маси являє собою придбання як комерційне, так і добровільне посилання.

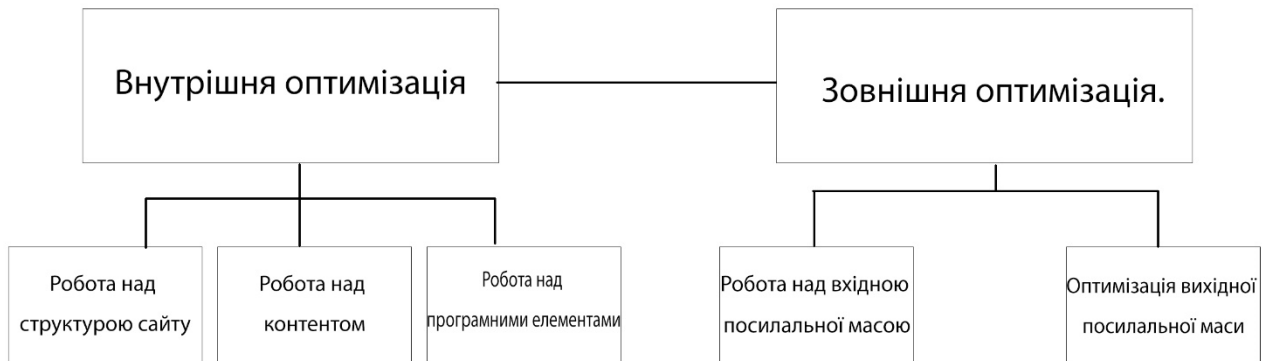


Рисунок 1.2 – Види пошукової оптимізації.

Не дивлячись на те, що внутрішня оптимізація стала більш важлива, ніж зовнішня, останній вид оптимізації також важливий для підвищення позиції інформаційно-управляючої системи. Зовнішня оптимізація впливає на такі показники, як ТИЦ і PR.

Крім того, існує певний поділ на типи пошукової оптимізації в залежності від застосовуваних методів, так звані: «біла» і «чорна» оптимізація.

Під «білою» оптимізацією розуміють певний комплекс заходів, спрямованих на підвищення позиції інформаційно-управляючої системи в пошуковій видачі, в результаті яких не порушуються правила, встановлені пошуковими сервісами для індексованих інформаційно-управляючих систем. Іншими словами, оптимізатор не повинен намагатися обдурити пошукову систему, змушуючи її, за допомогою певних прийомів, думати, ніби оптимізується сайт є краще як в плані технічного виконання, так і за наявною інформацією, ніж він є насправді.

«Чорна» оптимізація ж навпаки, ставить своєю метою саме обман пошукової системи, що часто тягне за собою відповідні наслідки з боку пошукової машини. Сайт, викритий в шахрайстві, може бути знижений в загальній пошуковій видачі, а то і зовсім виключений з нього, не кажучи вже про морально-етичних аспектах такого роду оптимізації. Адже фахівці, які обирають такий вид робіт з оптимізації, далеко не завжди піклуються про зручність і їх інформативності сайту безпосередньо для користувача, що в цілому негативно відбивається на загальній картині Інтернету. До чорної SEO-оптимізації відносяться наступні методи:

1. Клоакінг – тобто коли боти і відвідувачі бачать різний контент.
2. Посилальний чи пошуковий спам.
3. Використання невидимого для інтернет-користувачів тексту, який бачать пошукові машини.
4. Використання сторінок-дорвеїв для перегону трафіку. Дорвей — інструмент чорного SEO, сайт, створений для перенаправлення користувачів на інший інтернет-ресурс. Сайт зайвий з точки зору користувача, який часто навіть не має можливості розглянути його сторінки.

Сайти-дорвеї часто реєструють на безкоштовних хостингах. Текстовий вміст часто безглуздий, складається з обривків пропозицій з величезною кількістю ключових слів. Такі інформаційно-управляючі системи тільки засмічують мережу Інтернет. Основна мета дора - проіндексуватися і потрапити в результати видачі.

Видима легкість чорної пошукової оптимізації часто обманює бажаючих її використовувати, так як ефект від застосування даного виду короточасний. Для утримання позицій необхідно використовувати якісні, перевірені і правильні методи SEO, тобто які відносяться до білої оптимізації.

Незважаючи на те, що алгоритми пошукових машин регулярно ускладнюються, і пошукова оптимізація в цілому перетворюється в опрацювання сайту під правила пошукових систем, все ще можна зустріти приклади застосування шкідливих методів SEO-оптимізації. Найчастіше, чорні SEO використовуються для веб-ресурсів, які не розраховані на довготривале існування: ймовірність бану не критична, якщо мета зібрати швидкий трафік досягнута.

1.5 Поняття електронної комерції та її основні складові

Електронна комерція – це сфера економіки, яка застосовується для покупки та продажу товарів в мережі інтернет.

До електронних продаж відносяться наступні глобальні категорії:

- онлайн-торгівля;
- обмін даними в мережі інтернет;
- дистанційне банківське обслуговування;
- перекази грошей між користувачами.

На сьогоднішній день, практично кожна існуюча організація має в розпорядженні власну систему. Це може бути звичайна інформаційно-управляюча система з даними про організацію, послугами, контактною інформацією, або це може бути інтернет-магазин.

У сучасному світі все більше процесів стають автоматизованими, тому очевидно, ця сфера інтернет бізнесу, буде успішно розвиватися і далі.

1.6 Аналіз сучасних робіт

Серед сучасних публікацій на тему розробки системи обліку та продажу, потрібно приділити увагу таким статтям що присвячені аналізу сучасних веб-технологій, принципам клієнто-орієнтованості користувацького інтерфейсу, серверним технологіям розробки систем та електронній комерції.

В статті Кветкіна П.В., «Аналіз сучасного напрямку розвитку веб-технологій» [25], відображені основні напрямки розвитку технологій, що використовуються при розробці інтернет-систем. Розглянуто найбільш актуальні для користувача проблеми і методи боротьби з ними, що дозволяють розкрити можливості системи. Описуються найбільш важливі напрямки розвитку веб-технологій в сучасному світі: застосування мінімалізму в дизайні, технології HTML5, технології CSS3, паралакс-ефект, виведення вікна меню при розгляді контенту для зручності користувача, розробка актуального мобільного застосування, підвищення соціального захисту.

Виявлені основні тенденції розвитку веб-технологій, такі як застосування мінімалізму в дизайні, використання технологій HTML5 та CSS3, розробка актуального мобільного додатку для користувачу, підвищення соціального захисту. Описані основні переваги адаптивної сітки. Виявлені основні показники зручності використання системи.

В статті авторів Пасюков А.А. та Якимов А.С. «Порівняння продуктивності серверних мов програмування» [26], описано порівняння двох найпоширеніших серверних мов програмування PHP та Python. Для порівняння продуктивності мов, використовувалася серверна машина для тестування на операційній системі Debian 9.2 Linux x64. Тести проводилися на процесорі intel i5 2550k з частотою 3400 MHz і 16 гігабайти оперативної пам'яті. Щоб уникнути додаткових затримок при тестуванні було розгорнуто web-сервер і тестування запускалося на тій же машині.

В ході статті були описані тести продуктивності мов програмування PHP та Python. Було виявлено, що PHP після виходу версії 7.0 працює швидше при виконанні більшості завдань, але споживає більше ресурсів, що можна компенсувати більш потужним сервером. У плані зручності подальшої оптимізації та доопрацювання сайтів також виграв PHP в зв'язку з тим, що всі версії сумісні., Що не можна сказати про Python.

У статті автора Добровольский А.А., «Сучасні принципи клієнтоорієнтованості користувацького інтерфейсу» [27], описані проблеми систем які не оптимізовані під мобільні пристрої. Виявлено основні принципи юзабіліті призначеного для користувача інтерфейсу як інструмент ефективного просування і повернення користувачів.

Стаття автора Голуб А.Н., «Електронна комерція» [28], присвячена темі електронної комерції та електронного бізнесу. В статті описані основні поняття електронної комерції, виділені два основних напрямку електронного бізнесу. Описані основні можливості електронної комерції.

1.7 Формулювання проблеми дослідження

Основними проблемами при побудові систем обліку та продажу, стає:

- неправильно спроектована структура системи, через що користувач не може самостійно розібратися, як оформити замовлення, куди клікнути, щоб дізнатися умови доставки і т.д;

- неправильно обрані технології розробки системи, через що система виглядає застарілою і не залучає нових клієнтів, для того щоб система була привабливою, вона повинна відповідати усім новим стандартам;

- нехтування Seo оптимізацією, використання чорної Seo. Видима легкість чорної пошукової оптимізації часто обманює бажаючих її використовувати, так як ефект від застосування даного виду короткочасний. Для утримання позицій необхідно використовувати якісні, перевірені і правильні методи SEO, тобто які відносяться до білої оптимізації.

Для вирішення цих проблем, було сформульовано такі основні завдання:

- проаналізувати актуальні технології й методи розробки інформаційно-управляючих систем;

- порівняти основні технології розробки інформаційно-управляючих систем;

- практично реалізувати систему обліку та продажу транспортних засобів.

1.8 Висновки до розділу 1

Розробка інформаційно-управляючих систем – одна з найбільш затребуваних послуг в області інтернет-технологій. Послуги формування інформаційно-управляючих систем в цілому замовляють фірми, задіяні в різних областях бізнесу, індивідуальні підприємці.

Цілі, що стоять перед інформаційно-управляючими системами для організації:

1. Взаємодія з потенціальним споживачем.
2. Залучення нових покупців і партнерів.
3. Новини.
4. Організація товарів.
5. Мінімальні затрати.

Особливості проектування структури інформаційно-управляючої системи:

- Необхідно дотримуватися одноманітність елементів.
- Слід ухилятися створення схожих сторінок. Якщо однотипна інформація може бути поміщена на одній сторінці, не слід розробляти для неї окрему.

- Не створювати розділи які дублюють один одного. Якщо функціонал будь-якого блоку частково повторюється в другому, потрібно позбавитися його. Не потрібно зайвий раз плутати відвідувачів ресурсу.
- Простий і швидкий доступ до всіх найбільш важливих розділів з усіх сторінок.
- Проектування інформаційно-управляючої системи повинно включати складання карти основних сторінок.

На даний час вже існує декілька способів розробки веб-додатків:

1. Створення системи за допомогою мов програмування.
2. Конструктори.
3. Фреймоврки.
4. Використання CMS.

Виявлені особливості проектування структури інформаційно-управляючої системи, необхідно дотримуватися одноманітність елементів, ухилятися створення схожих сторінок. Якщо однотипна інформація може бути поміщена на одній сторінці, не слід розробляти для неї окрему. Не створювати розділи які дублюють один одного. Якщо функціонал будь-якого блоку частково повторюється в другому, потрібно позбавитися його. Не потрібно зайвий раз плутати відвідувачів ресурсу. Робити простий і швидкий доступ до всіх важливих розділів.

РОЗДІЛ 2

АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ ОБЛІКУ І ПРОДАЖУ

2.1 Аналіз типів й основних характеристик серверів

Сервер - це комп'ютер, який надає послуги іншим, приєднаним до нього комп'ютерам і виконує сервісні завдання без участі людини. Сервери можна класифікувати, як по класу вирішуваних завдань, так і за кількістю обслуговуваних клієнтів.

Найголовніша характеристика сервера - це його продуктивність, яка залежить від декількох параметрів [13]:

- від типу і продуктивності процесорів;
- від обсягу і типу оперативної пам'яті;
- від продуктивності дискової підсистеми.

Чим більше процесорів складають начинку сервера і чим більше ядер в кожному з них, тим більше потужність всієї мережі. В принципі, вибираючи конфігурацію сервера, потрібно обов'язково передбачити можливість розширення його через деякий час, якщо виникне потреба. Для цього потрібно подбати про наявність процесорів, пам'яті. Пристроїв, сумісних з уже наявними.

Друга важлива характеристика сервера - його керованість. Повинні бути забезпечені такі функції, як віддалений моніторинг і діагностика. Тобто, щоб сервером можна було керувати на відстані: включати і перезавантажувати, діагностувати і виправляти неполадки навіть у вимкненому стані (за умови, що він підключений до електричної мережі).

Перші дві характеристики: продуктивність і керованість - в значній мірі впливають на надійність сервера, що має на увазі не тільки фізичну його надійність і якісну збірку, але і програмну, яка складається в стабільній роботі всіх програм.

2.1.1 Основні типи серверів

Файл-сервер. Це централізоване сховище інформації, доступ до дисків якого мають підключені в локальну мережу персональні комп'ютери. Основне завдання файлового сервера зводиться до надійного збереження даних і безперебійному доступу до них, а в разі пошкодження файлів - повного їх відновлення.

Сервер бази даних. Засіб не стільки зберігання і доступу, скільки обробки масивів інформації. Через клієнтські запити запитувана інформація витягується, дані обробляються, структуруються, змінюються в залежності від налаштувань сервера. Керують роботою таких серверів СУБД, найвідоміші з них - MS SQL Server, Oracle, MySQL. Залежно від кількості користувачів і розміру бази даних, а також перспективи їх збільшення в майбутньому, визначають такі важливі характеристики сервера бази даних, як потужність і масштабованість.

Принт-сервер. Дозволяє використовувати один принтер для обслуговування декількох комп'ютерів. Функції принт-сервера - прийняти запити на висновок друку, вибудувати їх в чергу і згідно з нею відправляти на принтер. Таким чином, економляться кошти на комплектацію кожного комп'ютера власним принтером, їх пам'ять звільняється для інших завдань, раціонально використовується офісний простір.

Сервер робочої групи. Багатофункціональне апаратне рішення для групи комп'ютерів, як правило, не більше 20. Об'єднує в собі можливості файлового сервера, сервера додатків, бази даних, принт/факс-сервера, поштового та інших, в залежності від потреб. Зазвичай має один процесор, найчастіше використовується в невеликих фірмах, де немає потреби у виділенні серверів для окремих завдань.

Контролер домену. Головний комп'ютер в локальній мережі, що має ієрархічну структуру. Через контролер домену здійснюється централізоване управління ресурсами домену - обліковими записами комп'ютерів і користувачів. За допомогою служби директорій Active Directory він зберігає дані про користувачів і здійснює їх аутентифікацію для доступу до ресурсів локальної мережі. Працює під управлінням серверних ОС від MS Windows, починаючи з Windows 2000 Server. Контролер домену - важливий елемент мережевої інфраструктури великих компаній. Крім того, він може виконувати роль файлового сервера і сервера друку.

Поштовий сервер. Сервер повідомлень. Основне завдання такого сервера полягає в розпізнаванні адрес вхідних повідомлень електронної кореспонденції, та розподілі її по локальній мережі, відправку вихідної інформації, забезпечення внутрішнього інформаційного обміну. Поштовий сервер забезпечує надійну фільтрацію спаму і шкідливих програм, що розповсюджуються з повідомленнями, і захищає внутрішню інформацію від несанкціонованого доступу.

Сервер FTP. Невід'ємна частина технічного забезпечення Всесвітньої Павутини. Їх завдання - переміщати файли за запитом простих файлових менеджерів за допомогою стандартного протоколу File Transfer Protocol.

Проксі-сервер. Посередник між користувачами локальної мережі та Інтернетом. Забезпечує безпечний вихід в інтернет, захищаючи від небажаного доступу ззовні і при необхідності обмежуючи вихід на певні ресурси користувачам локальної мережі. Крім того,

виконує ряд інших функцій: облік і економія трафіку шляхом стиснення даних, кешування, анонімізація доступу.

Web-сервер. Спеціально виділений комп'ютер, який відповідає за доступ до сайту компанії користувачів Інтернету, коректне і швидке відображення статичних або динамічних сторінок. Веб-сервер зобов'язаний забезпечити безперебійну роботу Інтернет-ресурсу з урахуванням відвідуваності, протистояти мережевим атакам, не допускати можливості злому. В останні роки веб-сервером називають частіше не саму машину, а програму, що виконує перераховані функції.

2.1.2 Актуальні серверні рішення

Сервери Intel серії 1400E. Представник оновленої бюджетної серії серверів на базі Intel Xeon з шиною 800МГц і кеш-пам'яттю до 2Мб з підтримкою технології EM64T. Призначений для монтування в стійку. Сервери даної серії мають високу продуктивність і масштабованість, дисковою підсистемою на базі Ultra 320 SCSI або Serial ATA з апаратним RAID рівнів 0, 1, 5 з можливістю заміни дисків. Сфера застосування 1400E - WEB-хостинг, інтернет / інтернет-служби і термінальні служби.

Сервери Intel серії 2400A. Нова професійна серія серверів на базі Intel Xeon, з шиною 800МГц і кеш-пам'яттю до 2Мб з підтримкою технології EM64T. Форм-фактор 2U для монтування в стійку. Сервери сімейства мають високу продуктивність і масштабованість, дискову підсистему на базі Ultra SCSI 320 або Serial ATA з апаратним RAID рівнів 0, 1, 5, 10 і 50. Сфера застосування 2400A: принт-сервіси, електронна пошта і документообіг, корпоративна інфраструктура, WEB-сервери, контролери домену, малі бази даних.

Сервери Intel серії 5300E. Нова модель сервера, яка добре зарекомендувала себе з економічної серії серверів Intel. Сервер побудований на базі Intel Xeon з шиною 800МГц і кеш-пам'яттю до 2Мб з підтримкою технології EM64T. Сервери даного сімейства призначені для застосування в сфері малого та середнього бізнесу. Серія 5300E - це висока продуктивність і широкі можливості розширення при демократичній ціні. Ключові характеристики: частота системної шини 800MHz, двоканальна пам'ять DDR 333, дискова підсистема на базі Ultra 320 SCSI або Serial ATA з апаратним RAID рівнів 0, 1, 5, 10, 50 з можливістю заміни дисків, форм-фактор - вежа.

Сервери Intel серії 5300A. Нова професійна серія серверів на базі Intel Xeon, з шиною 800МГц і кеш-пам'яттю до 2Мб з підтримкою технології EM64T. Можливе виконання типу п'єдестал або для монтування в стійку. Сервери сімейства мають високу продуктивність і масштабованість, дисковою підсистемою на базі Ultra SCSI 320 або Serial ATA з апаратним

RAID рівнів 0, 1, 5, 10 і 50. Містка дискова підсистема до 10-ти дисків з гарячою заміною дозволяє побудувати RAID-масив будь-якої складності. Можливість резервування по харчуванню і дисковий масив з гарячою заміною дисків робить дану модель готовою до роботи з критичними додатками малого і середнього бізнесу. Сфера застосування 5300A: принт-сервіси, електронна пошта і документообіг, корпоративна інфраструктура, WEB-сервери, контролери домену, малі бази даних.

2.2 Клієнт-серверна взаємодія і ролі серверів

Комп'ютери які вміщуються до з'єднання єдиної-системи, ніколи не можуть бути рівноправними. Одні машини володіють ресурсами, а другі звертаються до цих ресурсів. Комп'ютер, або додаток які керують ресурсом, називають сервером. Клієнт та сервер можуть розташовуватися, як на одному комп'ютері, так і на різних комп'ютерах які пов'язані між собою інтернет павутиною.

В ході багаторівневого відображення обчислювальних систем, виділяють три класифікації положень обчислювальних систем:

1. Функції введення даних і їх відображення. Ця функція забезпечує взаємодію з користувачем;
2. Практичні функції, характерні для вирішення завдання;
3. Додатки керування доступу до ресурсів.

Виконання цих ролей забезпечується програмними інструментами, які можна представити у вигляді взаємопов'язаних компонентів (рис 2.2).



Рисунок 2.2 – Ролі мережевого додатку

- компонент подання даних відповідає за відображення інтерфейсу;
- практичний компонент реалізує вирішення існуючого завдання;
- компонент доступу до ресурсів забезпечує доступ ресурсів;

Мережа являє всі ці компоненти, але, в загальному випадку вони розподілені між вузлами. Завдання зводиться до забезпечення взаємодії між цими компонентами.

Архітектура «клієнт-сервер» визначає принципи роботи в Інтернеті, де є сервери, вузли-постачальники функцій і споживачі. Кожна технологія визначає положення взаємодії між клієнтом і сервером, які називають протокол обміну.

2.3 Порівняння веб-серверів Apache та Nginx

Apache - це HTTP-сервер, що має високу надійність і гнучкість. Під HTTP-сервером слід розуміти програмне забезпечення для обробки HTTP запитів. Основна робота Apache це обробка і відповідь на HTTP-запити і генерування динамічного вмісту.

Nginx - веб-сервер і поштовий проксі-сервер, що працює на Unix-подібних операційних системах.

Порівняння веб-серверів Apache та Nginx зазначені в табл. 2.1

Таблиця 2.1 – Порівняння веб-серверів Apache та Nginx

	Apache	Nginx
1. Метод обробки з'єднань з клієнтами	Apache на кожен запит від клієнта створює окремий процес. Виглядає це наступним чином - клієнт відправляє запит, веб-сервер створює окремий процес на цей запит, відповідає клієнту і блокує процес до тих пір, поки покупець не закрити з'єднання. Це легко і просто в реалізації, налагодженні і моніторингу, але, якщо великий проект, то коли процесів буде дуже багато, обробка з'єднань сповільнюється, пам'ять закінчується, CPU зростає.	Nginx складається з master-процесу і декількох дочірніх процесів. Майстер процес зазвичай один - він створює дочірні процеси, зчитує конфігурацію і відкриває порти. Воркер зазвичай кілька, розробники nginx радять кількість У nginx використовується нескінченний цикл, який біжить по всім сполуками і відповідає на запити клієнтів. Коли з'єднання закривається, воно видаляється. Це рішення ідеально підходить для проектів, які обслуговують 10 тисяч з'єднань одночасно. При цьому, завантаження CPU і використання пам'яті зазвичай рівномірні, без видимих піків.
2. Віддача контенту	Apache - може генерувати як статичний контент, так і динамічний.	Nginx - віддає тільки статистику і генерувати динамічний контент не вміє.

3. Конфігурація	Має можливості конфігурувати обробку з'єднань на рівні директорій. Робиться це за допомогою прихованого файлу .htaccess, що дозволяє налаштовувати права доступу, авторизацію, аутентифікацію, політику кешування. Це досить-таки зручне рішення для користувачів, тому що дозволяє змінювати конфігурацію на льоту, без перезавантаження сервера і без наявності доступу до основного конфігу сервера.	Nginx не підтримує конфігурацію на рівні каталогів. Існує один конфігураційний файл на весь проект, який обробляє master. Якщо потрібно оновити конфігурацію, то необхідно відправити сигнал SIGHUP майстру, який в свою чергу перезавантажує конфігурацію.
4.Робота з модулями	Apache за довгий час існування обзавівся близько 60 офіційними модулями, і ще більшим числом неофіційних. Модулі динамічно підключаються, не вимагають збірки і перезавантаження веб-сервера.	Nginx має близько 130 офіційних модулів. На відміну від Apache, модулі Nginx не можуть бути динамічно завантажені на льоту і вимагають складання. Це набагато складніше, але вважається безпечніше.
5.Інтерпретація запитів	Apache має можливість інтерпретувати запит як фізичний ресурс в файлової системі або як URI, який вимагає додаткової обробки.	Nginx, щоб працювати і в якості веб-сервера, і в якості проксі-сервера. З цієї причини він працює в першу чергу з URI, транслюючи їх при необхідності в запити до файлової системи.
6. Швидкість роботи	На основі тестів продуктивності, Nginx приблизно в 2.5 рази швидше віддає статичку, ніж Apache. Це досить-таки велика перевага. З точки зору пам'яті, обидва сервера використовують один і той же обсяг ресурсів	
7. Підтримка ОС	Apache прекрасно працює на Unix-подібних операційних системах, також розробники цього веб-сервера повністю підтримують лінійку Microsoft Windows, включаючи останні версії цієї ОС.	Nginx також підтримує роботу на безлічі Unix-подібних ОС і має деяку підтримку Windows, яка не є повною.

2.4 Аналіз актуальних методів розробки інформаційно-управляючої системи

Аналізуючи основні підходи до розробки інформаційно-управляючої системи, було вибрано засіб створення системи за допомогою мов програмування, так як цей спосіб найбільш привабливим у сфері веб-програмування. Завдяки цьому способу, можна досягти максимально привабливого результату для користувача та пошукових роботів.

Для того щоб зрозуміти як розробляється інформаційно-управляюча система, потрібно розібратися, із яких частин складається кожен веб-сайт [9]:

1. Графічний інтерфейс - це система графічних зображень, за допомогою яких користувач може "спілкуватися" з комп'ютером. Це вікна, іконки, панелі управління -

зображення, що дозволяють дати комп'ютеру команду за допомогою кліка. Для порівняння, протилежністю графічного інтерфейсу є текстовий інтерфейс, для роботи в якому користувач вводить спеціально сформульовані команди в командний рядок. Вся інформація при цьому постає в текстовому вигляді, наприклад, як при налаштуванні комп'ютера в BIOS.

2. Програмна частина - це сукупність програм, які забезпечують "взаєморозуміння" між комп'ютером і користувачем. Вони обробляють запити, що надходять, готують графічний інтерфейс і видають відповідь. Наприклад, якщо користувач хоче перейти по посиланню на іншу частину сайту, саме програмна частина аналізує суть запиту і відкриває нову сторінку.
3. Серверне обладнання - це комп'ютерне обладнання, оснащене спеціальним програмним забезпеченням для зберігання самої системи і даних користувачів, забезпечення безперервної доступності сайту в інтернеті і взаємодії зі сторонніми веб-службами.

2.5 Аналіз технологій розробки клієнтської частини системи

Верстка веб-сторінок - це створення структури html-коду, яка розміщує елементи веб-сторінки у вікні браузера, згідно з розробленим макетом, таким чином, щоб елементи дизайну виглядали аналогічно макету. Верстка веб-сторінок відрізняється від поліграфічної тим, що необхідно враховувати різницю відображення елементів в різних браузерах і різницю в розмірах робочого простору пристроїв.

Процес складний і має творчу основу, жоден із способів не є канонічним і прийнятим як основа. Всі підходи до верстки мають як переваги, так і недоліки. Хоча робота верстальника прихована від очей, саме вона забезпечує безперебійність при роботі на різних пристроях, а також швидкість завантаження кожної сторінки системи.

Різновиди верстки:

1. Фіксована верстка або статична. Незалежно від розмірів вікна браузера або пристрою, ширина сторінок буде постійною. Всі елементи займають строго певну ширину в пікселях на сторінці. Якщо дозвіл екрана великий і вікно браузера розгорнуто на весь екран, то, як правило, з боків залишається вільне місце. І навпаки, на мобільних пристроях при фіксованій верстці знизу на сторінці з'являється смуга прокрутки.

2. Гумова верстка. Сторінка займає всю ширину браузера, яких би розмірів вона не була і на якому б пристрою вона не була відкрита. Ширина елементів сторінки задається у відсотках від ширини вікна, і тому займають всю доступну область. При цій верстці важко домогтися

гарного зручного дизайну при всіх можливих дозволах екранів, адже сторінки будуть виглядати по-різному.

3. Таблична верстка або верстка таблицями. В даному випадку сітка сторінки будується за допомогою таблиць. На сторінці є головна таблиця, а в її осередках при необхідності розташовують вкладені таблиці, а в їх осередках можуть бути нові таблиці і так до нескінченності. Код виходить громіздким, це дуже незручно. Сьогодні так сторінки вже давно не верстають. Більш того, пошукові системи не люблять такі сторінки і погано їх індексують.

4. Блокова верстка, верстка блоками або div-верстка. Це, мабуть, найпоширеніша верстка сьогодні. Сітка сторінок конструюється з безлічі блоків <div>, які вкладені один в одного. Для доступу до них використовуються атрибути id і class.

5. Адаптивна верстка або мобільна верстка. Її ще іноді називають респонсивная верстка. Ключова особливість в тому, що сторінки добре адаптуються під будь-який дозвіл екрана користувача. Не важливо, відрили цю сторінку на стаціонарному ноутбучі, на здоровенному широкоформатному моніторі або на смартфоні - в будь-якому випадку сторінка повинна добре виглядати і бути зручною для користувача. Це досягається тим, що використовують кілька таблиць стилів під різні дозволи [40].

6. Гнучка верстка або flex верстка. Спочатку застосовується відома всім блокова верстка, а потім потрібні блоки перетворюють у флекс-контейнери. У стилях елемента вказують display: flex; Елементу можна вказати напрямок головної осі і вирівнювання. Ці можливості стали доступні з приходом CSS3. Тепер сторінки стають дуже гнучкими [32].

7. Семантична верстка. Вона з'явилася логічним продовженням блокової верстки і стала доступна в HTML5. Нові теги роблять сторінку більш структурованою. Пошуковики люблять такі сторінки.

8. Верста без помилок. Це верстка, виконана відповідно до стандартів W3C. Перевірити свою HTML-сторінку на коректність ви можете за допомогою спеціального валідатора W3C.

9. Кросбраузерна верстка. При такій верстці сторінки виглядають однаково в різних браузерах. Перше, з чого зазвичай починають - підключають до сторінці спеціальний CSS файл - скидання стилів.

2.5.1 Адаптивна верстка

Адаптивна верстка сайту дозволяє веб-сторінкам автоматично підлаштовуватися під екрани планшетів і смартфонів (рис 2.3). Мобільний інтернет-трафік зростає з кожним роком (рис 2.4) і щоб ефективно обробляти цей трафік, потрібно пропонувати користувачам адаптивні сайти зі зручним інтерфейсом.



Рисунок 2.3 – Адаптивний дизайн інтернет системи

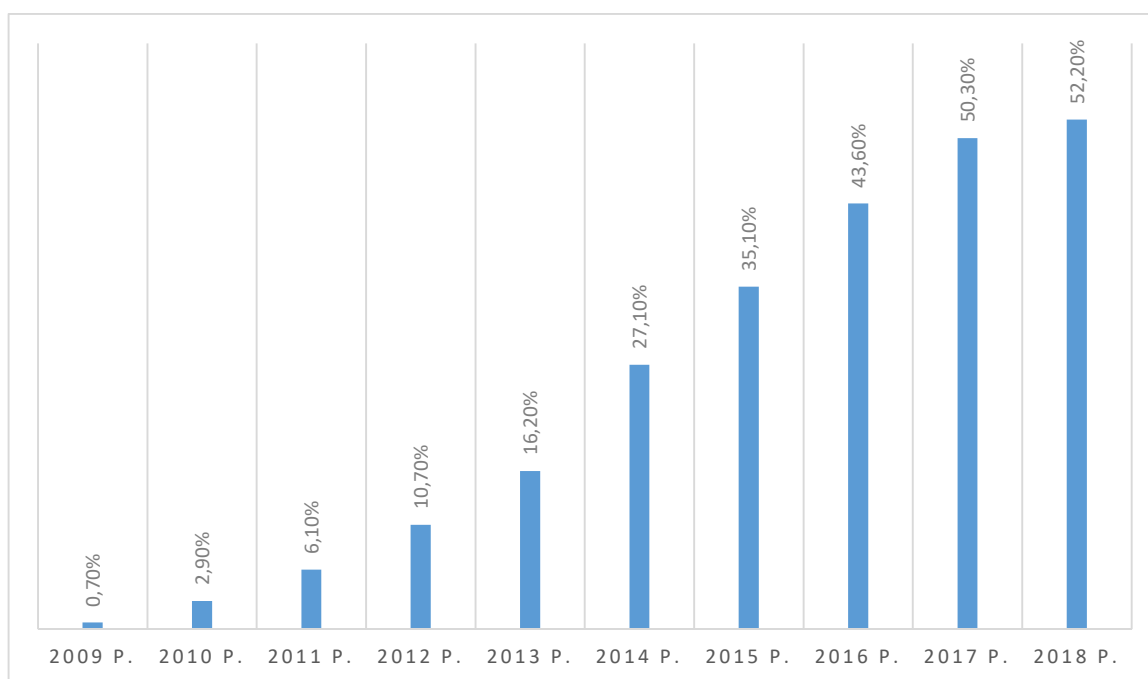


Рисунок 2.4 -Відсоток відкриття сайтів з мобільних пристроїв

Пошукові системи використовують ряд критеріїв для оцінки адаптивності сайту при перегляді на мобільних пристроях. Google намагається спростити користування Інтернетом для власників смартфонів і планшетів, відзначаючи в мобільній видачі адаптовані під мобільні пристрої сайти спеціальною позначкою mobile-friendly. В Яндексі також працює алгоритм, який віддає перевагу сайтам з мобільною версією для користувачів в мобільному пошуку.

Адаптивна верстка передбачає відсутність горизонтальної смуги прокрутки і масштабованих областей при перегляді на будь-якому пристрої, читабельний текст і великі

області для клікабельних елементів. За допомогою медіа-запитів можна управляти компонуванням і розташуванням блоків на сторінці, перебудовуючи шаблон таким чином, щоб він адаптувався під різні розміри екранів пристроїв [40].

Оптимізація під мобільні пристрої дозволяє:

- значно поліпшити використання інтернет-системи, і підвищується трафік користувачів;
- поліпшити розробку інтерфейсу системи, не потрібно створювати багато інтерфейсів під різні пристрої ;
- адаптувати систему при повороті екрану.

Порівняння адаптивної сітки та мобільної версії системи зазначені в табл. 2.2.

Таблиця 2.2 – Порівняння адаптивної сітки та мобільної версії системи

	Мобільна версія	Адаптивна сітка
Опис	Це створення окремої системи на піддомені, яка має власну верстку для мобільних пристроїв.	Це єдина верстка, яка гнучко підлаштовується під різні технічні пристрої. CSS забезпечує динамічну персоніфікацію макета під потрібний розмір екрану.
Переваги	<ol style="list-style-type: none"> 1. Мале споживання інтернет трафіку. 2. Можливість окремо просувати мобільну версію сайту. 	<ol style="list-style-type: none"> 1. Швидка швидкість завантаження, не вимагає перенаправлень на ту чи іншу версію. 2. Єдина адаптуєма версія кожної сторінки. 3. Зручність розробки та наповнення сторінки.
Недоліки	<ol style="list-style-type: none"> 1. Однакові сторінки можуть негативно відбитися на просуванні сайту. 2. Збільшення навантаження на пошукових роботів. 3. SEO просування потрібно реалізовувати окремо від основної системи. 4. Мобільна версія створюється окремо під кожен гаджет, це робить її створення технічно більш складною і дорогою. 	Для інформаційно насичених ресурсів буває проблематично розмістити на екрані мобільного пристрою весь вміст системи.

2.5.1.1 Технологія адаптивної сітки Bootstrap

Bootstrap - це CSS-фреймворк, який спочатку створювався для внутрішнього використання компанією «Twitter» з робочою назвою «Twitter Blueprint», але в підсумку був опублікований у відкритому доступі і став хорошим набором інструментів для front-end - розробки під назвою «Bootstrap» [12].

Переваги фреймворку Bootstrap:

1. Висока швидкість розробки макетів сторінок. Bootstrap містить величезний набір готових рішень і елементів.
2. Кросбраузерність і адаптивність. Всі елементи фреймворку адаптивні під всі пристрої і коректно відображаються у всіх сучасних браузерах.
3. Легкість у використанні. Навіть людина, що має базові знання HTML і CSS, може вільно створювати web-сторінки з використанням фреймворку.
4. Простота в навчанні. У Bootstrap дуже хороша документація з великою кількістю прикладів готового коду.

При верстці адаптивного класичного макета: шапка, основна частина, бічна колонка і нижня частина, для коректного відображення потрібно розрахувати ширину у відсотках кожного елемента і забезпечити обтікання. Якщо з шапкою і нижньою частиною все зрозуміло, в більшості випадків ширина буде 100%, то для основної частини контенту і бічної колонки може бути 70/30 або 85/25, але при зменшенні екрану нас це не влаштує, потрібно буде робити по 100% і скидати обтікання. Ось для таких цілей і потрібна сітка Bootstrap. Просто задаються класи для блоків, які вказують, яку ширину повинен займати елемент і як він буде відображатися на різних пристроях. Сітка функціонує як таблиця, в якій є свої ряди і стовпці, максимальна кількість стовпців 12.

Сітку можна робити всередині іншої сітки скільки завгодно. Якщо робити всі блоки сайту з використанням сітки, то самостійно писати медіа-запити для їх адаптивності взагалі не доведеться.

2.5.2 Мова гіпертекстової розмітки HTML5

HTML5 - це інструмент для упорядкування Web-контенту. Він призначений для спрощення Web-проекування і Web-розробки за рахунок мови розмітки, що забезпечує стандартизований і інтуїтивно зрозумілий інтерфейс. HTML5 надає розробнику кошти для секціонування і структуризації Web-сторінок, а також дозволяє створювати відокремлені

компоненти, які не тільки забезпечують логічну організацію Web-сайту, але і надають йому можливості синдикації [7].

Мова HTML5 реалізує підхід до проектування Web-систем, заснований на відображенні інформації, оскільки він втілює саму суть відображення інформації - поділ і маркування інформації для спрощення її використання і розуміння. Саме в цьому полягає величезна семантична та естетична цінність HTML5. HTML5 надає дизайнерам і розробникам всіх рівнів можливості для надання в публічний доступ буквально будь-якого контенту - від простих текстів до мультимедійно насичених інтерактивних матеріалів.

HTML5 надає ефективні інструменти для управління даними, для малювання, для відтворення відео та аудіоконтенту. HTML5 полегшує розробку крос-браузерних Web-додатків, а також додатків для мобільних пристроїв. HTML5 відноситься до числа технологій, які стимулюють розвиток мобільних сервісів на основі хмарних обчислень. Крім того, HTML5 сприяє підвищенню гнучкості - завдяки можливості створення вражаючих і інтерактивних Web-сайтів. Пропонує нові теги і вдосконалення, в числі яких такі: елегантна структура, органи управління формами, API-інтерфейси, мультимедійні функції, підтримка баз даних, істотно збільшена швидкість обробки.

Нові можливості HTML5:

- теги з описовими назвами, які точно вказують, для утримання якого контенту призначені ці теги;
- істотно покращено зберігання даних;
- засоби Web Worker для виконання фонових процесів;
- інтерфейс WebSocket для установки постійного з'єднання між додатком і сервером;
- покращений доступ до збережених даних;
- підвищена швидкість збереження і завантаження сторінок;
- підтримка CSS3 при управлінні призначеним для користувача інтерфейсом, що забезпечує контентну орієнтованість HTML5;
- покращена обробка форм в браузері;
- API-інтерфейс баз даних на основі SQL, що дозволяє застосовувати локальне сховище на стороні клієнта;
- теги canvas і video, що дозволяють додавати графічні та відеоматеріали без встановлення сторонніх модулів;
- специфікація API-інтерфейсу Geolocation, що використовує геолокаційні можливості смартфонів в інтересах задіяння хмарних сервісів і додатків для мобільних пристроїв;

- вдосконалені форми, що ослабляють потреби в завантаженні коду JavaScript, що забезпечує більш ефективний зв'язок між мобільними пристроями і серверами cloud-середовища.

Новий стандарт надав нові елементи для структурування і розмітки текстового вмісту. Нові елементи дозволили поліпшити структуру веб-сторінки, додавши смислове значення укладеним в них змісту. Основні семантичні елементи HTML5 зазначені в табл. 2.3.

Таблиця 2.3 – Семантичні елементи HTML5

Тег	Опис
<header>	Групує вступні і навігаційні елементи.
<nav>	Призначений для розробки блоку навігації.
<article>	Використовується для групування записів.
<section>	Елемент являє собою універсальний розділ документа. Групує тематичне вміст і зазвичай містить заголовок.
<aside>	Групує вміст.
<footer>	Являє собою нижній колонтитул. Зазвичай містить інформацію про автора статті, дані про копірайт і т.д.
<address>	Використовується для визначення контактної інформації автора.

2.5.3 Каскадні таблиці стилів CSS3

CSS (Cascading Style Sheets), або каскадні таблиці стилів, використовуються для опису зовнішнього вигляду документа, написаного мовою розмітки. Зазвичай CSS-стилі використовуються для створення і зміни стилю елементів веб-сторінок і призначених для користувача інтерфейсів, написаних на мовах HTML і XHTML, але також можуть бути застосовані до будь-якого виду XML-документа, в тому числі XML, SVG і XUL.

Каскадні таблиці стилів описують правила форматування елементів за допомогою властивостей і допустимих значень цих властивостей. Для кожного елемента можна використовувати обмежений набір властивостей, інші властивості не будуть чинити на нього ніякого впливу [41].

Оголошення стилю складається з двох частин: елемента веб-сторінки - селектора, і команди форматування - блоку оголошення. Селектор повідомляє браузеру, який саме елемент

форматувати, а в блоці оголошення перераховуються форматує команди - властивості і їх значення (рис. 2.5).



Рисунок 2.5 – Оголошення стилю в CSS3

2.5.3.1 Препроцесор SASS

Sass - це мова препроцесінга, яка компілюється в CSS. Цей препроцесор робить написання коду простіше, пропонуючи для цього зовсім небагато важелів. Він приводить стилі документа в хороший, структурований формат, на відміну від плоского CSS. Він задіє повторно використані методи, логічні вирази і пару вбудованих функцій типу маніпуляції з кольорами, математичні списки і списки параметрів [42].

Плюси застосування препроцесорів:

- можливість повторного використання змінних в таблиці стилів;
- функціональний синтаксис;
- можливість завантаження скомпільованих файлів у форматі CSS на web-сервер.

Переваги Sass перед іншими препроцесорами:

- повна сумісність з CSS. Sass повністю сумісний з усіма версіями CSS. Можливо будь-які доступні бібліотеки CSS;

- багата функціональність. Sass може похвалитися великою кількістю можливостей, ніж будь-яка інша мова розширення CSS.

- фреймворки. Є нескінченна кількість фреймворків, побудованих на Sass. Compass, Bourbon і Susy.

Технологія Sass - це ефективний інструмент для того, щоб істотно спростити роботу веб-розробників, які використовують CSS. Застосування цієї технології сприяє суттєвому зменшенню повторюваності коду.

Застосування Sass також забезпечує збільшення продуктивності роботи програмістів і дизайнерів. Тому вивчення даної технології дозволяє розширити можливості і потенціал будь-

якого фахівця, який працює в сфері веб-розробки. Завдяки цим знанням, на створення якісних професійних сайтів буде йти на порядок менше часу.

2.5.4 Мова програмування JavaScript

У створенні веб-систем використовується безліч спеціальних мов, як розмітки, так і веб-програмування, які часто комбінуються розробниками в рамках одного проекту. Кожен з них відрізняється різним рівнем складності, і одним з найбільш часто вживаних є JavaScript. Сучасний JavaScript - це безпечна мова програмування загального призначення. Вона не надає низькорівневих засобів роботи з пам'яттю, процесором, так як спочатку був орієнтований на браузері, в яких це не потрібно.

Це об'єктно-орієнтована, скриптова мова програмування, яка використовується для реалізації інтерактивних елементів на сторінці. За допомогою JavaScript можна значно розширити функціонал сайту, буквально «оживити» його, налагоджуючи таким чином зворотний зв'язок з відвідувачем.

Відмінність цієї мови програмування від ще одного поширеного - PHP - в тому, що він виконується на стороні клієнта, а не сервера. А якщо бути точніше - за допомогою браузера. Звідси виникають і деякі ризики щодо безпеки, але, в той же час, знижується навантаження на сервер.

У сучасній веб-розробці мову JavaScript використовують:

- для організації взаємодії з користувачами на веб-сторінці;
- для реалізації логіки поведінки елементів веб-сторінки;
- для додавання різних ефектів, особливо анімації і графічних, а також тих, що вимагають математичних обчислень;
- для обробки даних в HTML-формах;
- для взаємодії з різними HTML-елементами і CSS-стилями;

Підтримка JavaScript, є у всіх сучасних браузерах, але іноді вона може бути за замовчуванням неактивна, і тоді її потрібно включити. Браузер сам видасть в цьому випадку повідомлення, що для повноцінної взаємодії зі сторінкою сайту, потрібно включити підтримку скриптів.

2.6 Аналіз технологій розробки динамічних систем

Більшість великих інтернет-систем використовують програмування серверної частини, щоб динамічно відображати різні дані взяті з бази даних, розміщеної на сервері і відсилаються

клієнту. Динамічні системи можуть підсвічувати контент, більш відповідний перевагам і звичкам користувача [43].

Веб-браузери взаємодіють з веб-серверами за допомогою гіпертекстового транспортного протоколу HTTP. Коли користувач натискає на посилання на веб-сторінці, заповнює форму або запускає пошук, HTTP запит відправляється з вашого браузера на сервер. Запит включає в себе URL, який визначає отриманий ресурс, метод, який визначає необхідну дію й може включати додаткову інформацію, закодовану в параметрах URL.

Веб сервери очікують повідомлень з клієнтськими запитами, обробляють їх по прибуттю і відповідають веб-браузеру за допомогою відповідного HTTP повідомлення. Відповідь містить рядок стану, яка показує, чи був запит успішним, чи ні (рис 2.6).

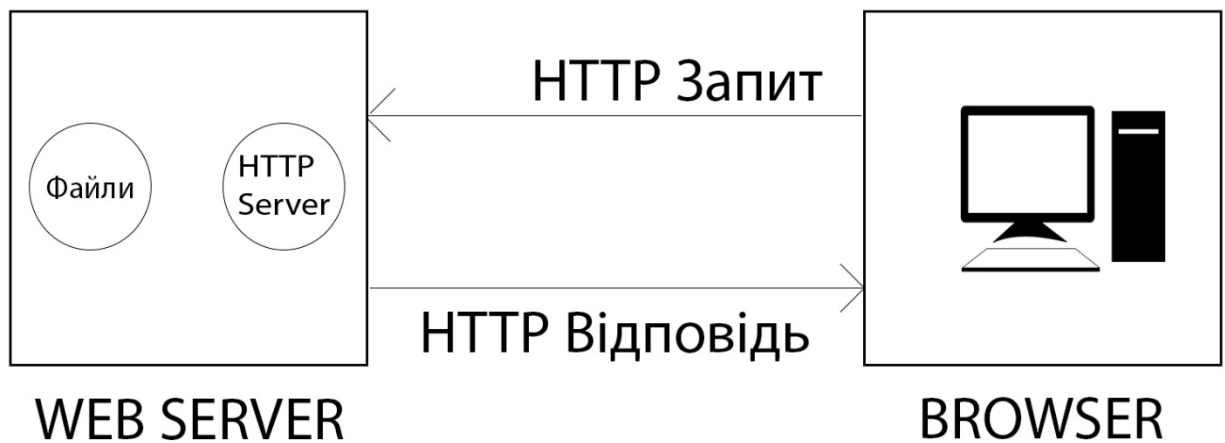


Рисунок 2.6 – Взаємодія веб-браузера і веб-сервера

Динамічна веб-система - це система, на якій у відповідь на запит, контент генерується динамічно тільки коли це потрібно. На динамічному веб-сайті HTML сторінки створюються отриманням даних з бази даних в плейсхолдери в HTML шаблонах, це набагато більш ефективний шлях зберігання великої кількості контенту, ніж використання статичних систем. Динамічна система може повертати різні дані в залежності від інформації, заснованої на URL, що поставляється користувачем або збережені переваги і може виробляти інші операції як частину відповіді.

Велика частина коду для підтримки динамічної веб-системи повинна запускатися на сервері. Створення цього коду відомо, як «програмування серверної частини».

Код серверної частини може бути написаний на будь-якій кількості мов програмування - приклади популярних мов серверної частини включають в себе PHP, Python, Ruby і C#. Код серверної частини має повний доступ до операційної системи сервера, і розробник може вибрати яку мову програмування він хотів би використовувати.

Розробники зазвичай пишуть свій код, використовуючи веб-фреймворки. Веб-фреймворки - це набір функцій, об'єктів, правил та інших конструкцій коду, спроектованих для вирішення типових завдань, прискорити розробку та спростити різні типи завдань, що виникають в конкретній галузі.

Застосування серверної частини:

- ефективно зберігання і доставка інформації;
- користувальницький досвід взаємодії з системою;
- контрольований доступ до контенту;
- зберігання інформації про стан системи;
- повідомлення та засоби зв'язку;
- аналіз даних системи о користувачах.

2.6.1 Мова серверних сценаріїв PHP

Мова PHP сьогодні використовується при розробці більшості сайтів і веб-додатків. Ця мова замислювалася авторами як простий і доступний інструмент для створення динамічних сторінок. Мова втілила в себе усі позитивні риси існуючих мов програмування для настільних систем, наприклад, багато функцій нагадують поведінку функцій з мови C або C++.

PHP - це свого роду деяка надбудова над системною мовою. Саме C здійснює аналіз і розбір коду PHP і переводить нескладні команди цієї мови в більш складні на C. Це спрощено, в реальності ж відбувається спочатку переклад коду PHP спочатку в опкоди, які аналізуються спеціальними функціями на C.

Весь написаний код на PHP, в момент запуску завантаження сторінки, завантажується в пам'ять сервера і виконується інтерпретація коду, його розбір. Якщо є помилки, робота зупиняється і видається помилка, якщо немає - код обробляється, перекладається в машинні коди і виконується на ПК.

Головна риса PHP - уявна простота і легкість розробки. З одного боку, це плюс, не потрібно стежити за типами змінних, не потрібно управляти виділенням пам'яті і виконувати інші рутинні операції. З іншого боку, така легкість породила безліч непрофесійно написаних скриптів, які можуть бути небезпечні, важкі в підтримці. Тобто, почати легко, але, щоб домогтися прийнятних результатів - потрібні роки досвіду. Добре написаний скрипт видно відразу, його відрізняє правильне оформлення коду, однакові відступи, узгоджене іменування змінних і функцій. Крім зовнішнього оформлення, будь то ООП або процедурний стиль, важлива сама логіка програми. Необхідна ретельна оптимізація написаного коду, перевірка його на безпеку.

Таким чином, розробка на PHP може бути якісною, навіть незважаючи на всю вільність і простоту використання мови. Тільки на відміну від системних мов, PHP не стежить і не вчить правильним манерам програмування, тому потрібно самостійно дотримуватися встановлених правил розробки згідно з документацією.

2.6.2 Патерн Model-View-Controller

Патерн Model-View-Controller, являє собою шаблон проектування архітектури програмного забезпечення, основним завданням якого є відділення функцій роботи з даними від їх подання (рис 2.7). Теоретично, грамотно спроектований MVC-додаток дозволить фронтенд і бекенд розробникам в ході роботи не втручатися в зони відповідальності один одного [44].

Хоча спочатку MVC був спроектований для розробки десктоп-додатків, він був адаптований для сучасних завдань і користується у веб-розробників величезною популярністю, оскільки за рахунок поділу відповідальності стало можливим створювати більш ясний, готовий до повторного використання код. Патерн MVC призводить до створення ясних, модульних систем, що дозволяє розробникам дуже швидко вносити зміни в існуючий код.

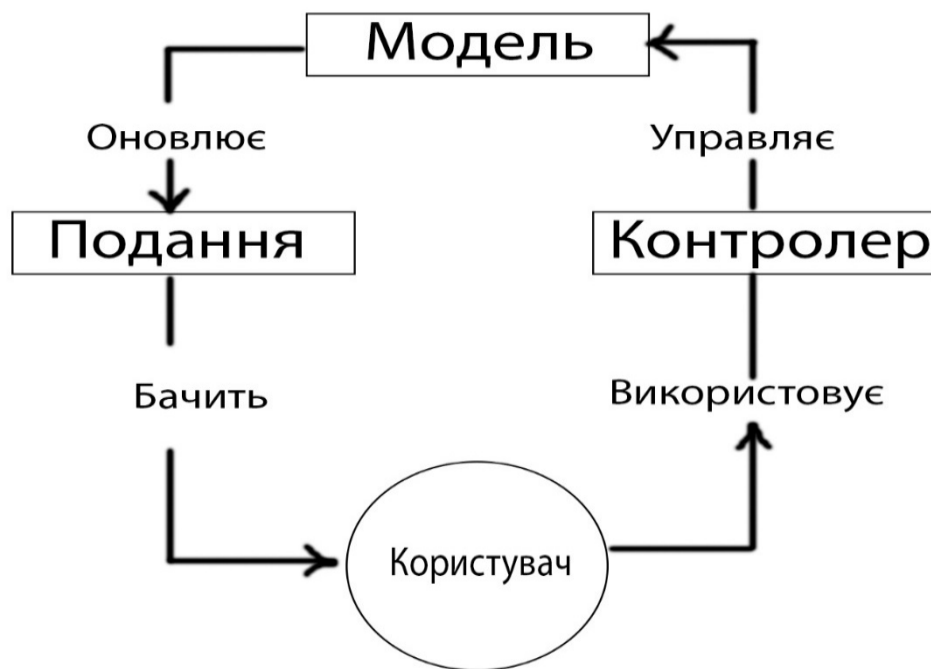


Рисунок 2.7 – Принцип роботи моделі MVC

Модель. Модель використовується для доступу і маніпулювання даними. У більшості випадків модель - це те, що використовується для доступу до сховища даних. Модель надає

інтерфейс для пошуку даних, їх створення, модифікації і видалення зі сховища. В контексті патерну MVC модель є посередником між поданням і контролером.

Подання. Подання - це те, де дані, отримані від моделі, виводяться в потрібному вигляді. У традиційних веб-додатках, розроблених в рамках MVC-патерну, уявлення - це частина системи, де виконується генерація HTML-коду. Подання також відповідає за отримання дій від користувача з тим щоб відправити їх контролеру. Наприклад, уявлення відображає кнопку в інтерфейсі, а після її натискання викликає відповідну дію контролера.

Контролер. Контролер - це остання частина зв'язки MVC. Завданням контролера є отримання даних від користувача і маніпуляція моделлю. Саме контролер взаємодіє з користувачем.

2.6.3 Система управління контентом

Система управління контентом (CMS) - це набір скриптів для створення, редагування і управління контентом сайту. Прикладами CMS є WordPress, Joomla, PrestaShop [45].

Якщо раніше більшість систем були статичними, і вимагали внесення правок в їх вміст вручну, то зараз динаміка розвитку проектів вимагає готовності швидко реагувати на зміни і впроваджувати їх з максимальною оперативністю. При цьому не всі користувачі хочуть або можуть собі дозволити звертатися до розробників, особливо якщо система вимагає постійної роботи над контентом.

У свою чергу, системи управління контентом дозволяють користувачам, які не володіють навичками розробки сайтів і знаннями мов програмування, самостійно працювати над створенням і зміною системи.

Суть роботи CMS укладена в схемі поділу вмісту системи і її дизайну. Користувачеві надається можливість вибрати шаблон - заготовку, якій заздалегідь визначено оформлення сторінки, і залишається тільки заповнити його потрібною інформацією

Як правило, CMS використовується для таких систем:

- блог, форум;
- інтернет-магазин;
- соціальні мережі;
- персональні системи;
- корпоративні системи;
- портали.

Тим не менше, більшість CMS гнучко налаштовуються і можуть бути використані для розробки систем різної спрямованості. Наприклад, найбільш популярним і універсальним варіантом є WordPress, на якому можливо створити практично будь-який проект.

Переваги CMS:

- просто і зручно використовувати;
- доступний, широкий функціонал;
- систему можна створити дуже швидко;

Недоліки CMS:

- популярні CMS уразливі;
- необхідно стежити за оновленням CMS;
- підвищене споживання ресурсів.

Існують також CMS, розроблені на замовлення під конкретний проект, як наслідок, їх функціонал не такий широкий, ніж у масових систем, але максимально відповідає поставленим завданням і не містить зайвих інструментів. На індивідуальних CMS створюються складні і масштабні системи зі специфічними особливостями.

2.6.3.1 Порівняння популярних систем управління контентом

Порівняння особливостей систем управління контролем зазначені в табл. 2.4.

Таблиця 2.4 – Порівняння особливостей CMS

	Drupal	Joomla	WordPress
Опис	Потужний інструмент, орієнтований на розробників і дозволяє створювати складні системи.	Компроміс Drupal і WordPress.	WordPress має велику кількість функцій, плагінів і віджетів.
Простота використання	Використовує широкі можливості при створенні систем. Від версії до версії спрощується і стає легше в засвоєнні	Значно простіше Drupal, але все одно складніше WordPress. Drupal не примхливий до розуміння структури та термінології, що дає можливість самостійно побудувати досить складну інформаційну систему не знаючи мов веб-програмування	WordPress найпростіша з CMS. За допомогою WordPress можна швидко створити просту систему. WordPress приваблює своєю легкістю в перенесенні тексту з документів в систему, чого не можуть Drupal

			та Joomla.
Особливості	Drupal відома своєю потужною систематизацією і можливостями по розмітці, категоризації та організації складного контенту	Joomla є платформою для спільнот з акцентом на соціальні функції	Зробивши систему і передавши її, можна не турбуватися про проблеми з подальшим наповненням її новим власником.
Модулі для кешування	Pressflow - несе в собі поліпшення, спрямовані на поліпшення продуктивності системи	JotCache оптимізує видачу поискових запитів.	WP Super Cache - оптимізує продуктивність системи, генеруючи статичний HTML-файли з одержуваного контенту з баз даних
Використання	Сучасні системи, великі спільноти, інтернет-магазини	Інтернет-магазини, соціальні мережі	Усі системи що вимагають можливості швидко додавати контент.

2.7 Аналіз положень індексування веб-системи

Одним з найважливіших понять в пошуковій оптимізації є індексування інформаційно-управляючої системи в пошукових системах. Індексція інформаційно-управляючої системи - це процес, що проводиться роботами пошукових систем, який полягає в дослідженні ними сторінок сайту з виявленням використовуваних слів, частоти їх вживання, із внесенням отриманої інформації в бази даних, які використовуються для формування відповідей на пошукові запити [46]. Роботи пошукової системи:

1. Основний – сканує контент веб-ресурсу і його окремих сторінок;
2. Швидкий – індексує тільки нову інформацію, яка була додана після оновлення.

При першому відвідуванні в базу потрапляють все нові системи, якщо вони підходять під вимоги пошукової системи. Під час повторного візиту інформація лише доповнюється деталями.

Чим швидше відбувається додавання сторінки в індекс, тим краще для веб-ресурсу. Однак пошукові роботи не можуть виконувати такий великий обсяг роботи так само часто, як оновлюється наповнення сайтів. Індексція в «Яндекс» в середньому займає одну-два тижні, а

Google – кілька днів. З метою прискорення індексації ресурсів, для яких дуже важливо швидке попадання інформації в базу, застосовується спеціальний робот, який відвідує такі інформаційно-управляючі системи від одного до декількох разів на день.

На швидкість індексації прямо впливають кілька факторів:

1. Відсутність помилок, що уповільнюють процес збору інформації пошуковим роботом.
2. Авторитетність ресурсу.
3. Частота оновлення контенту.
4. Частота додавання нового контенту.
5. Рівень вкладеності сторінок.
6. Коректно заповнений файл sitemap.xml.
7. Обмеження robots.txt.

Індексування є невід’ємною частиною роботи пошукових систем. База отримана в процесі сканування веб-сайту, використовується для формування нагальній видачі.

Індексування інформаційно-управляючої системи дуже складний процес. Щоб управляти їм, потрібно використовувати файл robots.txt і такі приписи, як Allow, Disallow, Crawl-delay, User-agent та інші.

Robots.txt — це стандарт виключень для роботів. Файл robots.txt використовується для часткового управління індексуванням веб-ресурсу пошуковими роботами. Цей файл складається з набору інструкцій для пошукових машин, за допомогою яких можна задати файли, сторінки або каталоги сайту, які не мають індексуватися.

Файл robots.txt може використовуватися для вказівки розташування файлу і може показати, що саме потрібно проіндексувати пошуковому роботу в першу чергу.

Sitemap.xml – це файл, в якому містяться посилання на всі сторінки мережевого ресурсу. Такий файл називають картою інформаційно-управляючої системи. Його створюють перед тим, як дозволити пошуковим системам провести першу індексацію контенту. У процесі розвитку інформаційно-управляючої системи цей файл оновлюють вручну або в автоматичному режимі.

Основні функції карти інформаційно-управляючої системи:

1. Зберігання актуальної інформації про індексованих сторінках.
2. Передача пошуковим системам інформації про частоту оновлення контенту.
3. Передача роботу вказівок з приводу індексації, включаючи встановлений пріоритет сторінок відносно один одного і в структурі інформаційно-управляючої системи.
4. Надання посилань на весь індексований контент в зручному форматі, що забезпечує швидкий доступ до всіх сторінок мережевого ресурсу.

Файл Sitemap.xml створюють після складання семантичного ядра, визначення структури інформаційно-управляючої системи і підготовки до просування в інтернеті. Його розміщують в директорії ресурсу, після чого повідомляють пошуковим системам його мережеву адресу.

Основні особливості складання sitemap:

1. Карта інформаційно-управляючої системи sitemap.xml повинна знаходитися в директорії домену сайту. Не можна складати карту одночасно для доменів та піддоменів, прописуючи посилання усіх сторінок в одному файлі.

2. Ім'я для карти інформаційно-управляючої системи можна вибрати будь-яку, головне, не помилитися з розширенням файлу.

3. Максимальна кількість посилань – 50 тис.

4. Максимальний обсяг файлу – 10 Мб.

5. Якщо потрібно зробити карту інформаційно-управляючої системи з кількістю посилань, що перевищує максимальне значення, то дозволяється ділити файл на декілька штук. Те ж саме роблять при перевищенні обсягу.

6. Інформацію у файлі зберігають у кодуванні UTF-8. Якщо сайту присвоєно кириличний домен, перетворення робити не обов'язково. Пошукові системи розпізнають адреси в будь-якому вигляді.

При SEO-оптимізації створення карти інформаційно-управляючої системи є важливим процесом, від правильності проведення якого залежать швидкість і ефективність майбутнього індексування вмісту. Якщо власну мапу можна швидко створити вручну, просто скопіювавши в текстовий файл посилання на всі важливі сторінки, то з формуванням карти у форматі XML доведеться повозитися. У такому файлі інформація представлена у зручному для пошукових систем вигляді. Крім звичайних посилань на сторінки з контентом в ньому присутні спеціальні теги, директиви, параметри. Щоб правильно створити карту, вебмастер повинен навчитися користуватися ними.

2.8 Аналіз роботи популярних пошукових систем

На зорі появи Інтернету лише невелика кількість користувачів мала можливість до нього підключитися. Обсяг доступної інформації теж був невеликий. В Інтернеті в той час працювали в основному співробітники науково-дослідних організацій. Пошук інформації в Мережі було не так популярний, як в даний час.

Першою спробою організувати доступ до електронних даних через Інтернет зробила пошукова система Yahoo. Розробники компанії створили відкритий каталог сайтів, посилання на які були згруповані за відповідною тематикою. Із зростанням кількості ресурсів в базі

виникла необхідність додавання опції пошуку по каталогу. Це ще не була пошукова система в звичному для нас вигляді, вона шукала інформацію по внутрішній базі сайтів, а не по всім існуючим інтернет-ресурсам. Подібні каталоги посилань користувалися раніше великою популярністю, але сьогодні втратили свою актуальність, оскільки кількість інформаційно-управляючих систем постійно збільшується.

Основним поняттям, з яким працює будь-яка пошукова система, є пошуковий індекс — певна структура даних, що відображає інформацію про документи і розташування в них ключових слів.

Принципи роботи багатьох пошукових систем дуже схожі. Основна відмінність полягає у підході до методу упорядкування ресурсів в пошуковій видачі.

Для збільшення швидкості роботи пошукової системи архітектура пошуку складається з двох елементів:

- базовий пошук;
- метапошук.

Базовий пошук — програма, яка шукає в рамках своєї частини індексу і видає користувачеві всі посилання, що відповідають пошуковому запиту.

Метапошук — програма, яка при обробці запиту визначає місцезнаходження користувача і видає готову пошукову видачу, якщо ключове слово популярне, якщо ж подібних запитів раніше не було, то в роботу підключається базовий пошук, який методом машинного навчання обробляє наявні в базі посилання та видає їх список.

Пошукова система одночасно аналізує користувача і безпосередньо пошуковий запит за такими критеріями:

- довжина;
- чіткість;
- популярність;
- конкурентність;
- синтаксис;
- географія.

Після того як пошуковий запит розібраний по параметрам і класифікований по одному з перерахованих типів, пошукова система підбирає функцію ранжирування.

Виділяють наступні типи запиту:

- навігаційний;
- інформаційний;
- транзакційний;
- мультимедійний;

- загальний;
- службовий.

В Україні, як і раніше, пошукова система Google займає перші позиції серед пошукових систем і впевнено тримає першу позицію (рис.2.8). А ось рейтинг системи Яндекс, який почав падати більше року тому, продовжує своє зниження. В даний момент на вересень 2018 року немає ніяких підстав припускати, що популярність цієї пошукової системи почне збільшуватися.

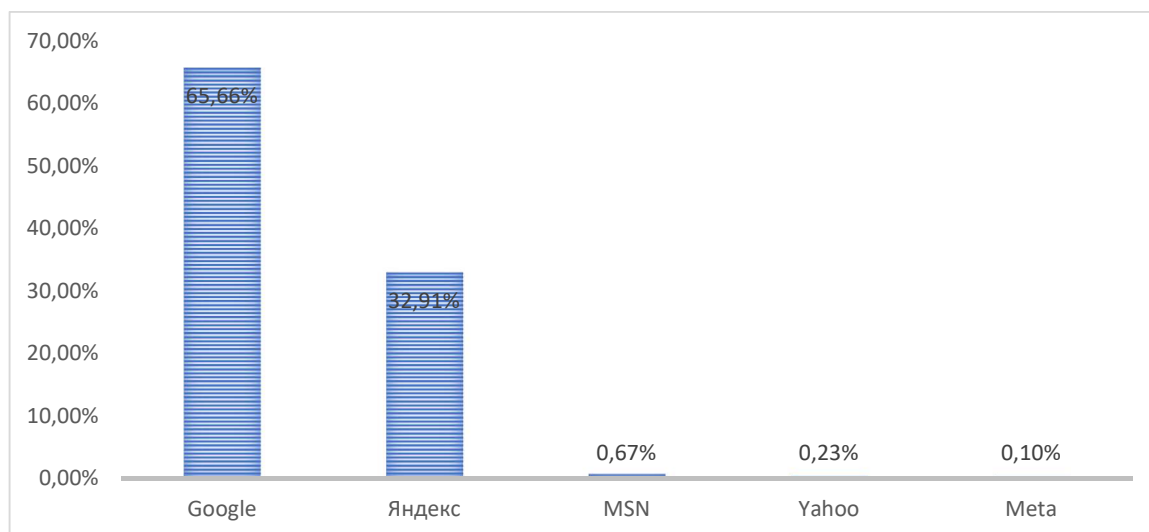


Рисунок 2.8 – Рейтинг пошукових систем в Україні на вересень 2018 року.

2.9 Висновки до розділу 2

У другому розділі дипломного проекту, були проаналізовані основні типи й характеристики серверів, проаналізована клієнт-серверна взаємодія. Получено актуальні методи розробки інформаційно-управляючої систем.

На основі аналізу була вибрана адаптивна верстка як можливість оптимізації системи під різні пристрої що дасть можливість:

- Значно поліпшити використання інтернет системи, і підвищити трафік користувачів.
- Поліпшити розробку інтерфейсу системи, не потрібно створювати багато інтерфейсів під різні пристрої.
- Адаптувати систему при повороті екрану.

Для реалізації адаптивної сітки, була обрана технологія Bootstrap, яка має такі особливості як:

1. Висока швидкість розробки макетів сторінок.

2. Кросбраузерність і адаптивність.

3. Легкість у використанні.

Були проаналізовані основні технології розробки клієнтської та серверної частини інформаційно-управляючої системи.

Проаналізовані положення індексування веб-системи пошуковими роботами й отримано особливості які впливають на індексацію:

1. Відсутність помилок, що уповільнюють процес збору інформації пошуковим роботом.
2. Авторитетність ресурсу.
3. Частота оновлення контенту.
4. Частота додавання нового контенту.
5. Рівень вкладеності сторінок.
6. Коректно заповнений файл sitemap.xml.
7. Обмеження robots.txt.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ З ОБЛІКУ І ПРОДАЖУ ТРАНСПОРТНИХ ЗАСОБІВ

3.1 Проектування архітектури системи обліку і продажу

Практична реалізація системи обліку і продажу починається з проектування архітектури. Архітектура системи - це систематизація інформації та навігації по ній з метою допомогти відвідувачам більш успішно знаходити потрібні їм дані. Добре продумана грамотна архітектура гарантує, що користувачі витратять менше часу на пошук потрібної інформації.

Розробка архітектури повинна вестися з урахуванням найбільш важливої інформації з точки зору просування товарів на інтернет-ринку.

У процесі створення структури нової системи, або оптимізації структури вже існуючої, необхідно концентрувати увагу споживачів саме на цій інформації і управляти відвідуваністю системи потенційними клієнтами саме в найбільш важливих розділах системи відповідно до позиціонування на ринку.

Грамотний розподіл пріоритетів між розділами і сторінками, зробить їх основними точками входу до системи, що дозволить потенційному споживачу швидко знайти необхідну йому інформацію про шуканий товар і послуги, що підвищить успішність бізнесу в інтернеті.

Архітектура інтернет-магазину повинна бути проста і інтуїтивно зручна (рис 3.1).

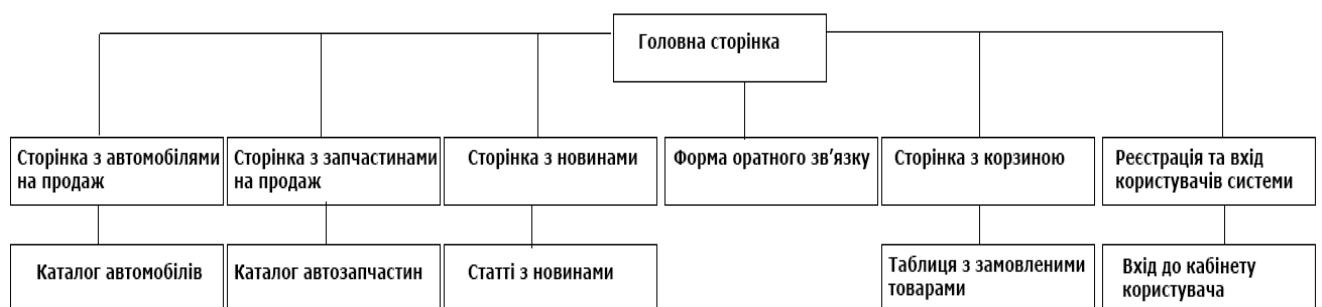


Рисунок 3.1 – Архітектура системи обліку та продажу транспортних засобів

Клієнтська частина буде реалізована за допомогою мови гіпертекстової розмітки HTML5, таблиць каскадного стилю CSS3, технології адаптивної сітки Bootstrap5 та мови програмування JavaScript.

Серверна частина буде реалізована за допомогою мови програмування Php й використовуючи модель поєднання клієнтської та серверної частини Model View Controller.

Система працюватиме з базою даних Microsoft MySQL (рис 3.2).

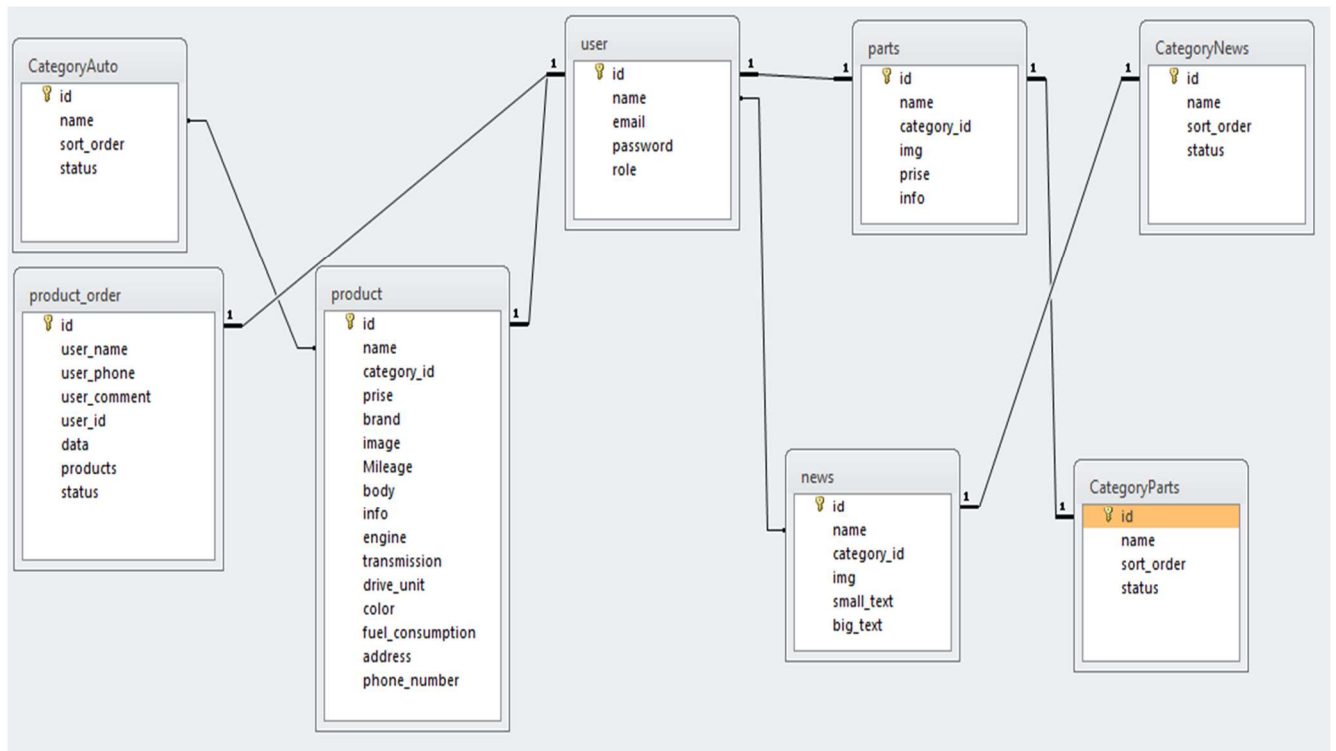


Рисунок 3.2 – Структура бази даних системи обліку і продажу

3.2 Розробка дизайну користувацької частини системи

Перше, що побачить користувач при вході до системи, це буде дизайн системи. Тому він повинен бути максимально зручним і простим у використанні.

Верстальщикам завжди доводиться вирішувати проблему адаптивності дизайну інтерфейсу до різних розмірів екрану. Є можливість створення безлічі копій дизайну під різні пристрої, але це не зручно, трудозатратно, для того, щоб зробити систему максимально зручною для всіх розмірів екранів, буде використовуватися адаптивна верстка, яка значно поліпшить використання інтернет системи, і підвищить трафік користувачів, спростить розробку інтерфейсу системи, буде адаптувати систему при повороті екрану.

Використання адаптивної системи має свої плюси над іншими методами розробки систем:

1. Адаптивна версія, а також її підтримка, коштує набагато менше, ніж створення окремої мобільної версії. Крім того, це нескладний і досить швидкий процес дизайну системи;
2. У адаптивної версії і сайту один url, тому не потрібно буде витратити час на редіректи, а клієнтам запам'ятовувати дві різних адреси сайту;
3. Гнучкість системи - навіть якщо багато інформації на сайті, вона буде виглядати красиво на будь-якому пристрої;

4. Для адаптивної версії існують свої вимоги Google. Якщо сайт буде правильно адаптований і оптимізований, показники істотно підуть вгору і система збільшиться у пошуковій видачі.

На смартфоні усі блоки системи, будуть розміщуватися один під одним, ховатимуться непотрібні для користувача смартфона блоки, а на планшетах, та екранах комп'ютерів, система буде приймати зовсім інші розміри і буде видно більше інформації на екрані. Результат розробки адаптивного інтерфейсу системи показаний на рис. 3.3 – 3.4.



Рисунок 3.3 – Відображення характеристик автомобіля на мобільному пристрої

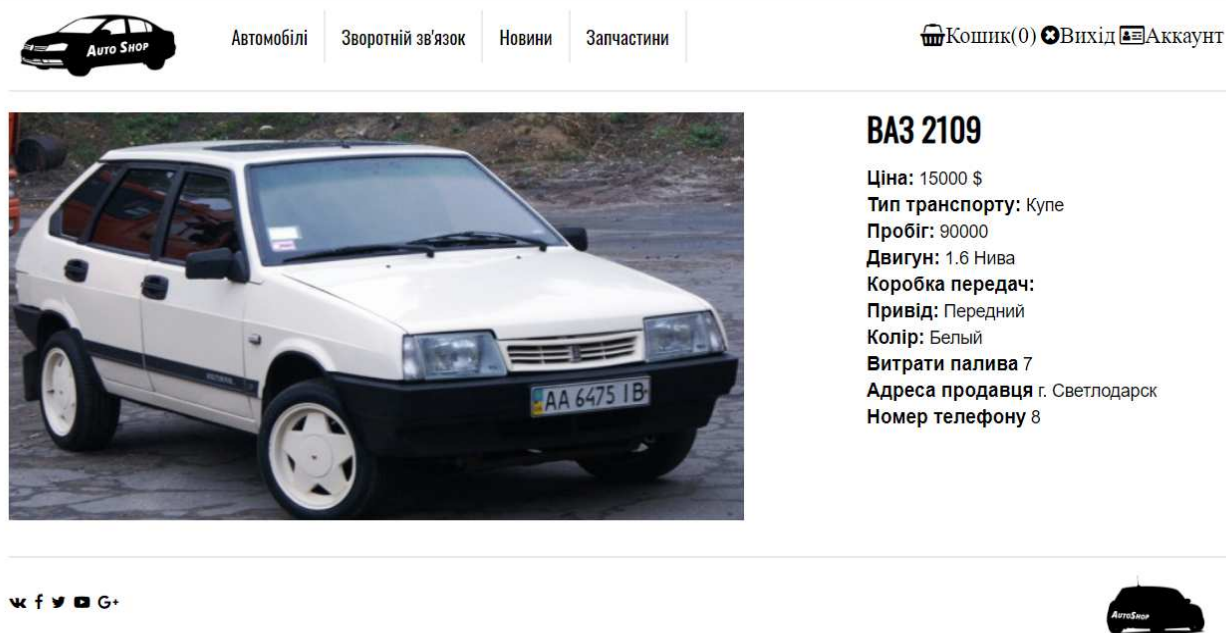


Рисунок 3.4 – Відображення характеристик автомобіля на екрані настільного комп'ютера

Для розробки адаптивного дизайну, використовується фреймворк Bootstrap5, який прискорить і спростить швидкість розробки макетів сторінок системи обліку і продажу транспортних засобів й зробить систему максимально адаптивною й зручною для використання користувачем.

Після розробки й реалізації адаптивного дизайну, було проведено тест системи за допомогою Mobile-Friendly-Test, в результаті перевірки було дано позитивну відповідь (рис 3.5)

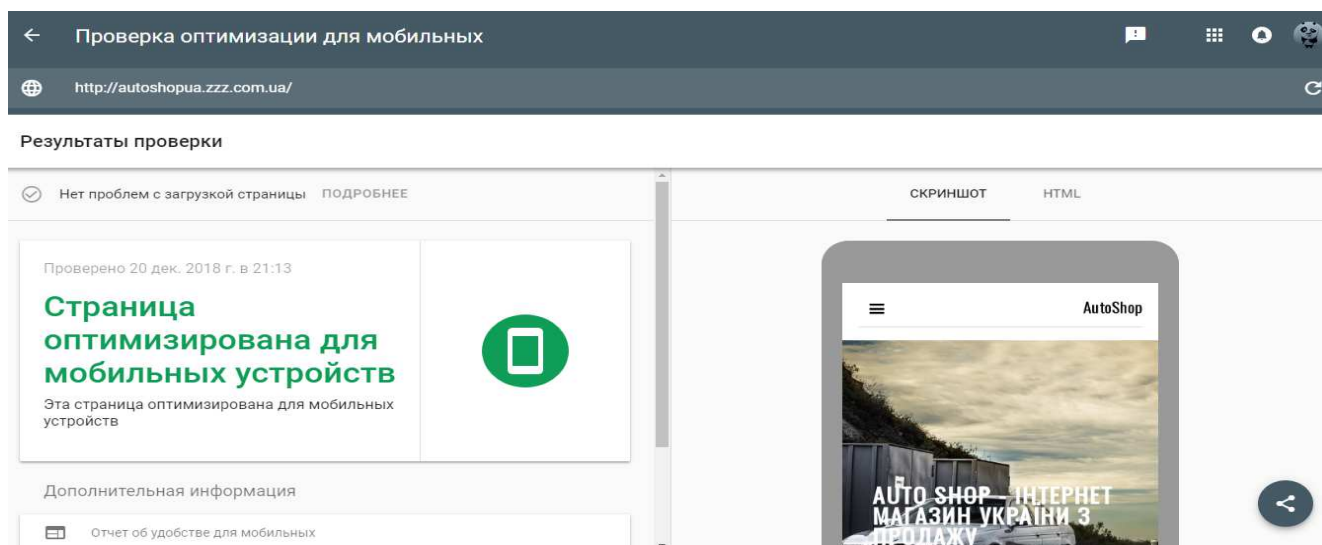


Рисунок 3.5 – Перевірка адаптивності системи за допомогою Mobile-Friendly-Test

Для того щоб система була максимально привабливою для користувачів, до дизайну було додано можливість зміни мови на англійську (рис3.6).

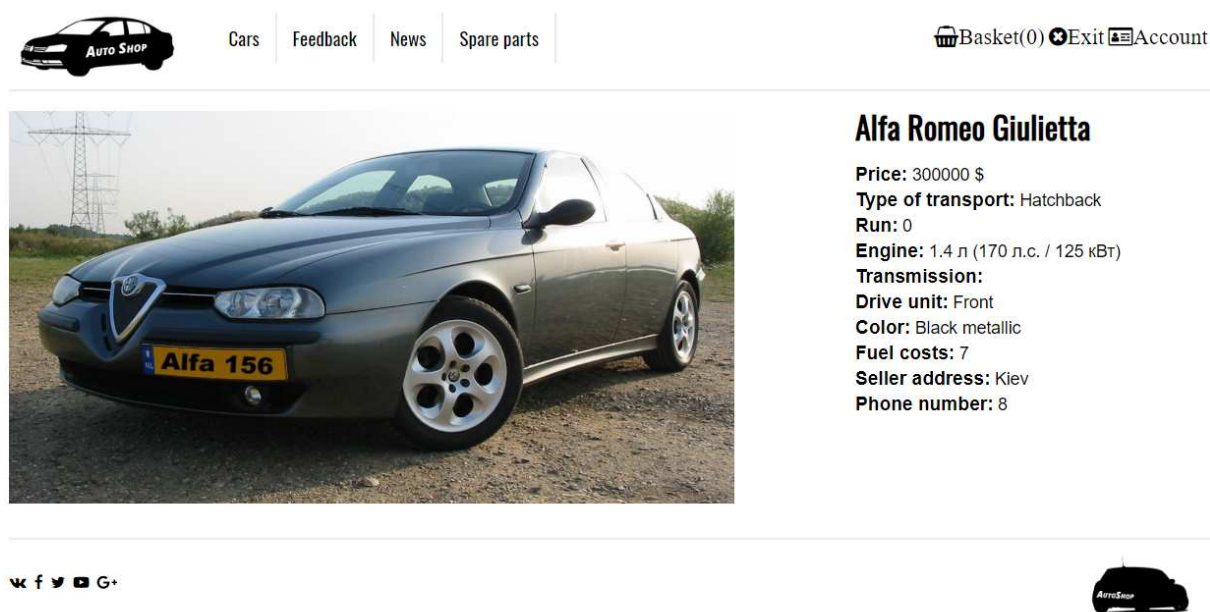


Рисунок 3.6 – Зміна мови в системі

3.3 Розробка та реалізація головної сторінки

Головна сторінка, це перше що побачить новий користувач системи, тому вона потрібна бути простою і лаконічною, щоб користувач не захотів відразу закрити сторінку. Головна сторінка повинна містити основні відомості про систему й меню сайту, за допомогою якого користувач зможе продовжити подальше використання системи. Результат розробки головної сторінки показаний на рис. 3.7.

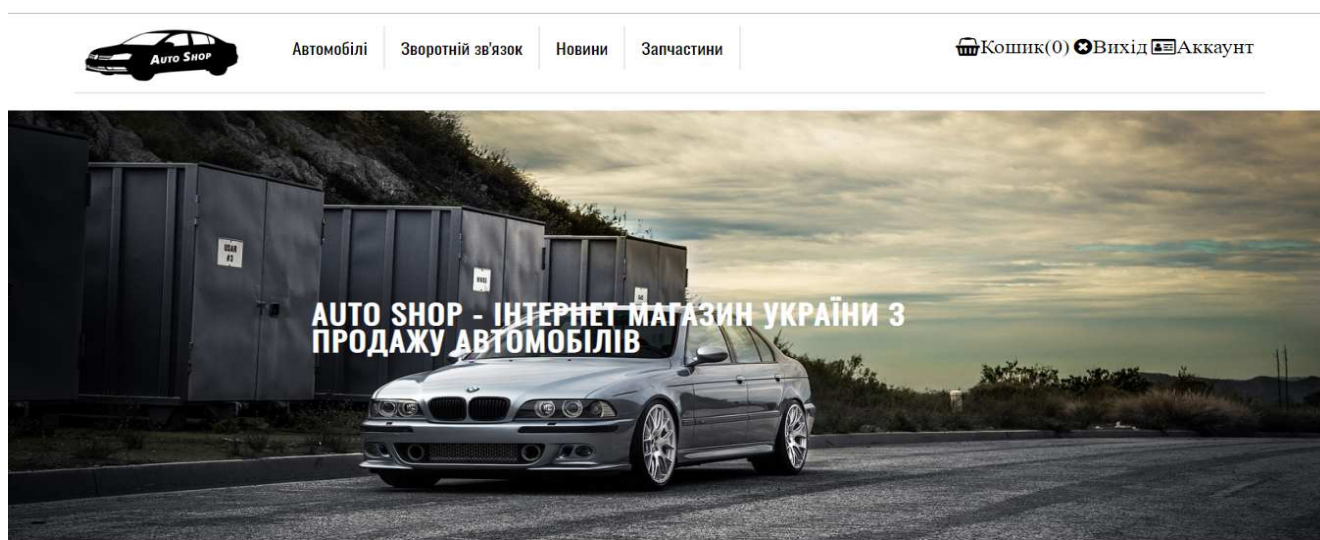


Рисунок 3.7 – Головна сторінка системи обліку і продажу транспортних засобів

Для зручності подальшого використання системи, додано меню з основними сторінками системи. Кожний підпункт меню системи, адресує користувача на відповідну сторінку:

- “Автомобілі”;
- “Зворотний зв’язок”;
- “Новини”;
- “Запчастини”.

Також додано можливість реєстрації, та можливість входу до аккаунту системи з головної сторінки. Це облегшить використання системи, користувачу не потрібно шукати якісь можливості системи, уся основна інформація додана до шапки (рис 3.8).

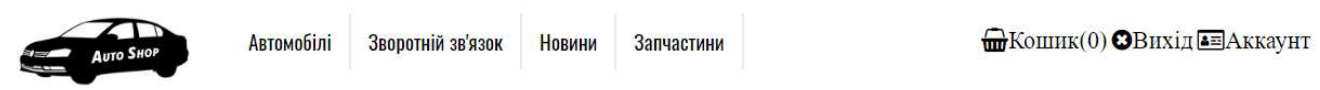


Рисунок 3.8 – Меню системи з основними сторінками

3.4 Розробка серверної частини системи

Для серверної частини використовуватиметься динамічне відображення інформації. Так як сторінок з автомобілями може бути досить багато і вони будуть постійно додаватимуться, потрібно щоб система сама генерувала на сервері нові сторінки без втручання розробника. Для того щоб це реалізувати, потрібно створити шаблон, за допомогою якого подальше будуть створюватися нові сторінки в системі, інформація яка буде автоматично генеруватися і заповнювати сторінки системи, буде зберігатися в базі даних.

Коли користувач запрошуватиме сторінку, відповідна інформація буде витягтися з бази, й вставлятися в шаблон, утворюючи нову web-сторінку, і відправлятися web-сервером в призначений для користувача браузер, який і буде відображати її.

Для того щоб робити динамічний веб-сервіс на комп'ютері, потрібно встановити локальний сервер Open Server. Open Server являє собою WAMP комплекс:

1. Windows - операційна система, для роботи в якій призначений даний локальний сервер.
2. Apache - web-сервер, який «піднімається» при запуску програми Open Server.
3. MySQL - дуже популярна система управління базами даних, яка є обов'язковою умовою для роботи багатьох двигунів сайтів.
4. PHP - інтерпретатор серверної мови програмування, на якому написано більшість CMS і створено безліч інших веб-додатків.

Apache надає багато можливостей, що дозволяють налаштувати Web-сервер відповідно до потреб індивідуальних і корпоративних користувачів. Apache дозволяє створювати віртуальні Web-вузли, а також виконує функції проху-сервера.

Далі потрібно вирішити, як зберігати інформаційне наповнення системи, яке відображається на Web-сторінці. У даній системі обліку буде створено базу даних в СУБД MySQL, яка дозволить розбити Web-контент на таблиці, що містять поля і записи з даними. СУБД MySQL - відмінний вибір для створення такої бази даних, внаслідок простоти у використанні і адмініструванні, вільної поширеності для різних платформ, включаючи Linux і Windows, і швидко зростаючій популярності.

В пакеті Open Sever є веб-додаток phpMyAdmin, який допомагає в зручній графічній формі без введення SQL запитів, управляти базами даних на сервері MySQL. Залежно від рівня доступу користувача, за допомогою цього серверного скрипта можна управляти як окремими БД, так і цілим сервером MySQL.

Першою і найбільш важливою дією при створенні бази даних є розробка її моделі, необхідно вибрати назву бази даних й визначити, що саме будуть містити таблиці бази даних.

Для системи продажу та обліку транспортних засобів, було створено базу даних під назвою «AutoShopUa», яка містить в собі вісім таблиць: «categoryAuto», «categoryNews», «categoryParts», «news», «parts», «product», «product_order», «user» (рис 3.9).

The screenshot shows the phpMyAdmin interface for the 'AutoShopUa' database. The left sidebar shows the database structure with tables: categoryAuto, categoryNews, categoryParts, news, parts, product, product_order, and user. The main window displays a table of database tables with the following data:

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
categoryAuto		9	InnoDB	utf8_general_ci	16 КИБ	-
categoryNews		0	InnoDB	utf8_general_ci	16 КИБ	-
categoryParts		0	InnoDB	utf8_general_ci	16 КИБ	-
news		0	InnoDB	utf8_general_ci	16 КИБ	-
parts		0	InnoDB	utf8_general_ci	16 КИБ	-
product		4	InnoDB	utf8_general_ci	16 КИБ	-
product_order		0	InnoDB	utf8_general_ci	16 КИБ	-
user		1	InnoDB	utf8_general_ci	16 КИБ	-
8 таблиц	Всего	14	InnoDB	utf8_general_ci	128 КИБ	0 Байт

Рисунок 3.9 - Вміст бази даних

Для того щоб система обліку мала можливість взаємодіяти з базою даних, було створено файл “db_params.php”, який підключить базу даних до системи (рис 3.10).

```

1  <?php
2
3      return array(
4          'host' => 'localhost',
5          'dbname' => 'AutoShopUa',
6          'user' => 'root',
7          'password' => "",
8      );

```

Рисунок 3.10 – Підключення бази даних до системи обліку та продажу транспортних засобів

Для розробки шаблону використовуватиметься мова гіпертекстової розмітки HTML5. А для того щоб система взаємодіяла з базою даних і шаблоном, буде використовуватиметься мова програмування PHP (рис 3.11).

```

<?php foreach ($categoriesParts as $categoryPart): ?>
    <li><a href="/categoryPart/<?php echo $categoryPart['id'];?>"
        <?php echo $categoryPart['name'];?>
    </a>
    </li>
<?php endforeach; ?>

```

Рисунок 3.11 – Приклад додання нових категорій на сторінку за допомогою PHP

Для того щоб з'єднати серверну і клієнтську частину, буде використовуватися модель уявлення Model-View-Controller:

- Модель надає дані і реагує на команди контролера, змінюючи свій стан.
- Подання відповідає за відображення даних моделі користувачеві, реагуючи на зміни моделі.
- Контролер інтерпретує дії користувача, сповіщаючи модель про необхідність змін.

3.5 Розробка системи реєстрації та входу

Для того щоб користувач мав можливість купувати автомобілі та запчастини в системі, потрібно розробити можливість реєстрації та авторизації покупців в системі. Також авторизація в системі допоможе розпізнати роль користувача на сайті й надати йому права адміністратора, якщо він ними володіє. Структура системи авторизації користувачів в системі зображена на рис. 3.12.

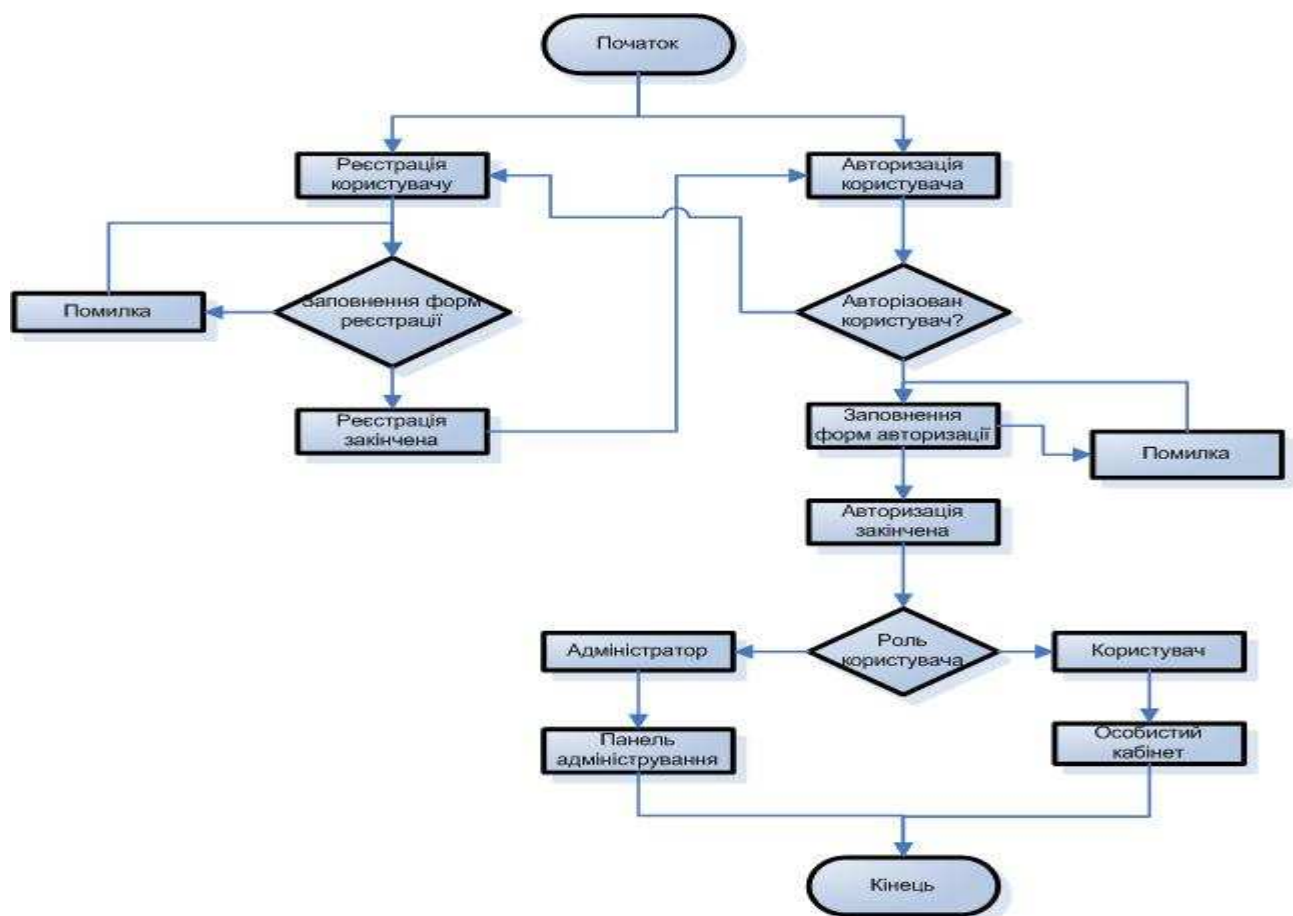


Рисунок 3.12 – Структура системи реєстрації-авторизації користувачів

Вхід до системи можливий тільки після того, як пройдена реєстрація. Один раз досить зареєструватися, а потім можна багаторазово заходити за допомогою опції “Війти”.

Після реєстрація користувача в системі:

- створюється аккаунт користувача;
- створюється обліковий запис;
- якщо користувач володіє правами адміністратора, то відкривається адмінпанель.

Для того щоб зареєструватися в системі, потрібно вибрати пункт меню “Реєстрація”, після чого відкриється нова сторінка з формою реєстрації (рис 3.13).

Рисунок 3.13 – Форма реєстрації в системі

Реєстрація в системі реалізована в файлі «UserController.php», за допомогою PHP5.4. Після того, як користувач заповнить усі поля реєстрація та натисне кнопку «Зареєструватися», усі данні з форми потраплять до функції «ActionRegister», яка перевірить заповнені поля на помилки і якщо помилок немає, то відправить змінні з даними до файлу «User.php» в функцію «register», де за допомогою PHP та SQL запитів усі данні будуть відправлені у таблицю «user». Реалізація функцій реєстрації «ActionRegister», «register» показані на рис. 3.14 – 3.15.

```
public static function register($name, $email, $password)
{
    $db = Db::getConnection();

    $sql = 'INSERT INTO user (name, email, password) '
        . 'VALUES (:name, :email, :password)';

    $result = $db->prepare($sql);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':email', $email, PDO::PARAM_STR);
    $result->bindParam(':password', $password, PDO::PARAM_STR);

    return $result->execute();
}
```

Рисунок 3.14 – Реалізація функції «register»


```

11 public function actionRegister()
12 {
13     // Переменные для формы
14     $name = false;
15     $email = false;
16     $password = false;
17     $result = false;
18
19     // Обработка формы
20     if (isset($_POST['submit'])) {
21         // Если форма отправлена
22         // Получаем данные из формы
23         $name = $_POST['name'];
24         $email = $_POST['email'];
25         $password = $_POST['password'];
26
27         // Флаг ошибок
28         $errors = false;
29
30         // Валидация полей
31         if (!User::checkName($name)) {
32             $errors[] = 'Имя не должно быть короче 2-х символов';
33         }
34         if (!User::checkEmail($email)) {
35             $errors[] = 'Неправильный email';
36         }
37         if (!User::checkPassword($password)) {
38             $errors[] = 'Пароль не должен быть короче 6-ти символов';
39         }
40         if (User::checkEmailExists($email)) {
41             $errors[] = 'Такой email уже используется';
42         }
43
44         if ($errors == false) {
45             // Если ошибок нет
46             // Регистрируем пользователя
47             $result = User::register($name, $email, $password);
48         }
49     }
50
51     // Подключаем вид
52     require_once(ROOT . '/views/user/register.php');
53     return true;
54 }

```

Рисунок 3.15 – Реалізація функції «ActionRegister»

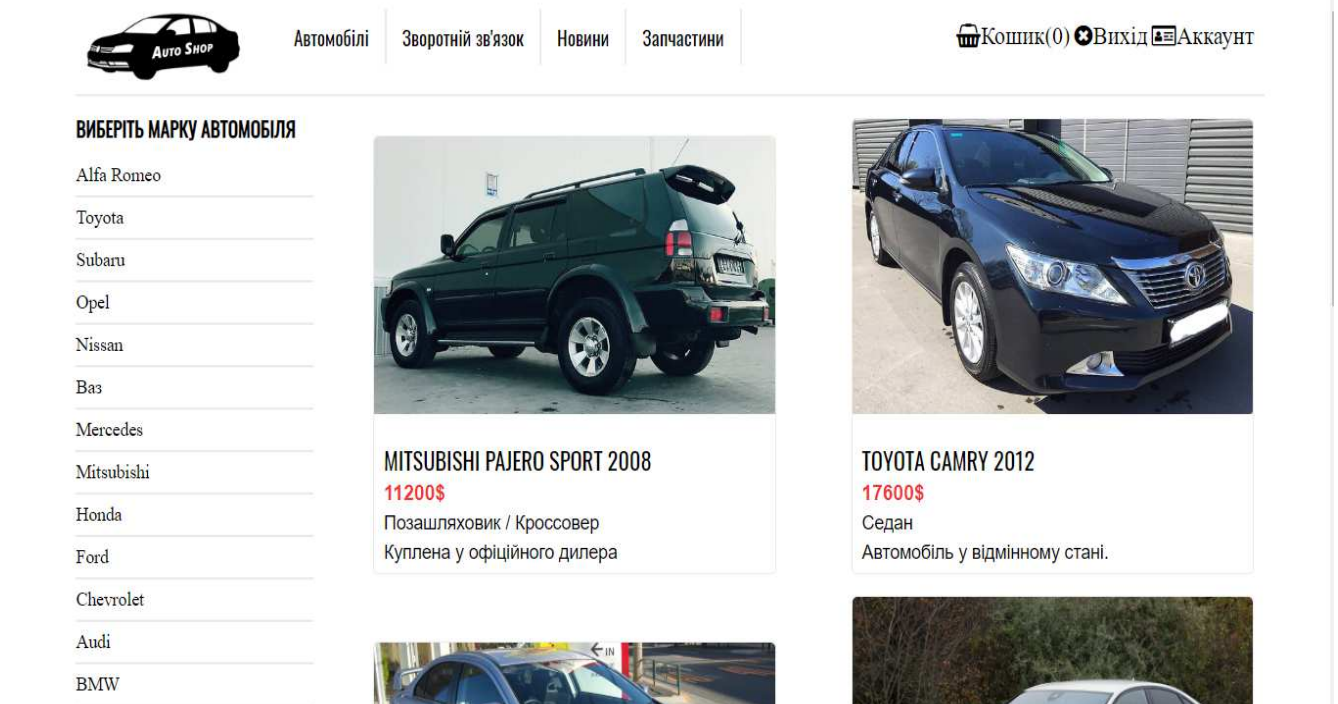
Після того, як користувач зареєструється в системі, йому буде виданий особистий кабінет, де він зможе редагувати данні свого профіля. Також після реєстрації користувача, в таблицю бази даних MySQL буде доданий новий користувач. За допомогою цієї таблиці, адміністратор системи має можливість видалити, або додати права адміністратора користувачу (рис 3.16).

				id	name	email	password	role
<input type="checkbox"/>				1	cola	cola@yandex.ru	12345	admin
<input type="checkbox"/>				2	Сергей	serega@gmail.com	161995s28	
<input type="checkbox"/>				3	Александр	moneycase@mail.ru	123456789	

Рисунок 3.16 – Таблица даних користувачів системи

3.6 Реалізації сторінки продажу транспорту

Основною і головною функцією системи, є продаж автомобілів, як б/у, так і нових. Для того щоб користувачу, було зручно користуватися системою й шукати автомобілі, було створено окрему сторінку “Автомобілі”. Де є можливість переглянути усі пропозиції автомобілів на продаж, або вибрати окремо марку автомобіля і переглянути тільки ті пропозиції авто які потрібні покупцеві. Реалізації сторінки з автомобілями показана на рис. 3.15.



Автомобілі | Зворотній зв'язок | Новини | Запчастини

Кошик(0) | Вихід | Аккаунт

ВИБЕРІТЬ МАРКУ АВТОМОБІЛЯ

- Alfa Romeo
- Toyota
- Subaru
- Opel
- Nissan
- Vaz
- Mercedes
- Mitsubishi
- Honda
- Ford
- Chevrolet
- Audi
- BMW

MITSUBISHI PAJERO SPORT 2008
11200\$
 Позашляховик / Кроссовер
 Куплена у офіційного дилера

TOYOTA CAMRY 2012
17600\$
 Седан
 Автомобіль у відмінному стані.

Рисунок 3.15 – Сторінка з вибором автомобілів на продаж

Якщо користувачу буде потрібно знайти автомобіль якимось відповідним параметрам, то в системі реалізований пошук по параметрам: «Тип кузова», «Ціна», «Тип палива». Це значно полегшить користування системою, не потрібно постійно шукати автомобілі з списку, достатньо задати потрібні параметри в пошуку. Реалізація пошуку показана на рис. 3.16.

Параметри автомобіля!

Тип кузова:

Ціна:

Тип палива:

Рисунок 3.16 – Пошук автомобілів в системі

Висновок товарів на сторінку виконується за допомогою файлу «Product.php», та функції «getLatestProducts», яка за допомогою SQL запитів буде брати товари з таблиці «product» й виводити товари на сторінку. На першу сторінку буде виведено шість останніх доданих авто в

систему, а далі авто будуть розподілені на інші сторінки. Реалізація функції додавання авто на сторінку та таблицю «product» показано на рис. 3.17 – 3.18.

```

const SHOW_BY_DEFAULT = 6;

/*получение массива с последними товарами*/
public static function getLatestProducts($count = self::SHOW_BY_DEFAULT)
{
    $count = intval($count);
    $db = Db::getConnection();
    $productsList = array();

    $result = $db->query('SELECT id, name, prise, image, body, info FROM product ORDER BY id DESC LIMIT ' . $count);

    $i = 0;
    while ($row = $result->fetch()) {
        $productsList[$i]['id'] = $row['id'];
        $productsList[$i]['name'] = $row['name'];
        $productsList[$i]['prise'] = $row['prise'];
        $productsList[$i]['image'] = $row['image'];
        $productsList[$i]['body'] = $row['body'];
        $productsList[$i]['info'] = $row['info'];
        $i++;
    }

    return $productsList;
}

```

Рисунок 3.17 – Реалізація функції додавання авто на сторінку

	id	name	category_id	prise	brand	image	Mileage	body	info	engine	transmission	drive_unit	color
<input type="checkbox"/>	1	BA3 2109	9	20000	BA3		150000	Купе	Не бита, не крашена	1.6 Нива		Передний	Серебро
<input type="checkbox"/>	2	BA3 2111	9	26000	BA3		130000	Универсал	Авто в хорошем состоянии.	1.6 Газ/ Бензин		Передний	Зеленые водоросли
<input type="checkbox"/>	3	BA3 2109	9	15000	BA3		90000	Купе	Авто в хорошем состоянии.	1.6 Нива		Передний	Белый
<input type="checkbox"/>	4	Alfa Romeo Giulietta	1	300000	Alfa Romeo		0	Hatchback	New car.	1.4 л (170 л.с. / 125 кВт)		Front	Black metallic
<input type="checkbox"/>	5	Audi A6 3.0 2011	3	25000	Audi		105000	Седан	Куплена у офіційного дилера	3 л. Бензин		Повний	Білий
<input type="checkbox"/>	6	Mitsubishi Lancer X 2008	7	7700	Mitsubishi		98000	Седан	Продам свій автомобіль	1.5 л., Бензин		Передний	Сірий метал
<input type="checkbox"/>	7	Toyota Camry 2012	13	17600	Toyota		145000	Седан	Автомобіль у відмінному стані.	2.5 л., Бензин		Передний	Чорний
<input type="checkbox"/>	8	Mitsubishi Pajero Sport 2008	7	11200	Mitsubishi		98000	Позашляховик / Кроссовер	Куплена у офіційного дилера	3 л., Газ / бензин		Повний	Чорний

Рисунок 3.18 – Таблиця з автомобілями в системі

Для того щоб користувач мав можливість подивитися усю інформацію про цікавий автомобіль, є можливість відкрити розширену інформацію про кожне авто, де можна буде дізнатися про:

- «Ціна»;
- «Вид транспорту»;

- «Пробіг»;
- «Двигун»;
- «Трансмiсія»;
- «Привідний пристрій»;
- «Колір»;
- «Витрати на паливо»;
- «Адреса продавця»;
- «Номер телефону».

Реалізація сторінки з розширеною інформацією про автомобіль показана на рис. 3.19.

Автомобілі | Зворотній зв'язок | Новини | Запчастини

Кошик(0) | Реєстрація | Вхід

Mitsubishi Pajero Sport 2008

Ціна: 11200 \$
Вид транспорту: Позашляховик / Кроссовер
Пробіг: 98000
Двигун: 3 л., Газ / бензин
Трансмiсія:
Привідний пристрій: Повний
Колір: Чорний
Витрати на паливо: 10
Адреса продавця: г. Сєверодонецьк
Номер телефону: 8

vk f G+

Рисунок 3.19 – Сторінка з розширеною інформацією

3.7 Реалізація сторінки управління системою

Найскладніше в реалізації системи обліку і продажу транспорту, є створення панелі адміністрування. Так як потрібно зібрати всю технічну складову системи і об'єднати в єдину систему управління

Для управління системою, було додано сторінку “Адміністрування”, вона дасть керуючому можливість редагувати усі сторінки без входження в бази даних, все буде здійснюватися за допомогою адмін-панелі.

Це не тільки зручніше с точки зору керування, а й надійніше, адміністратору не потрібно заходити в phpMyAdmin й шукати потрібну таблицю, або на сам сервер й шукати потрібний

файл, уся потрібна інформація буде показана в адмін-панелі, через яку є можливість додати, видалити, змінити інформацію.

Для того щоб користувач мав можливість потрапити до панелі адміністрування, потрібно щоб йому були видані права адміністратора, які виставляються в таблиці “user” бази даних.

Адмін панель в системі має такі пункти меню:

- «Автомобілі»;
- «Категорії автомобілів»;
- «Новини»;
- «Категорії новин»;
- «Запчастини»;
- «Категорії запчастин»;
- «Замовлення».

Реалізація панелі адміністратора показана на рис. 3.20.

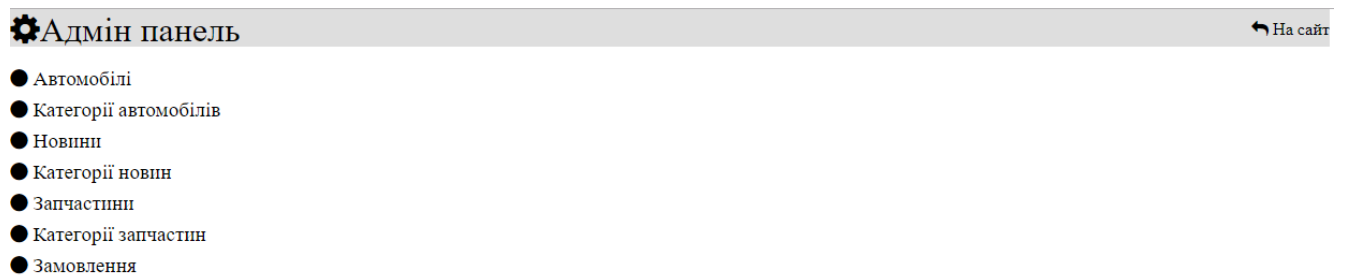


Рисунок 3.20 – Панель адміністрування системою продажу транспорту

Кожний з підпунктів меню панелі адміністрування, відповідає певній сторінці системи й певній таблиці в базі даних, це в декілька разів облегшує керування системою.

Так для того щоб додати новий автомобіль до бази даних, не потрібно заходити в phpMyAdmin, шукати потрібну таблицю й потім щось туди додавати, все що потрібно, це вибрати пункт меню «Автомобілі», де відкриється сторінка з таблицею додавання, видалення або редагування вже існуючих автомобілів в системі. Сторінка адміністрування автомобілів показана на рис. 3.21.

Адмін панель Керування автомобілями

Список товарів

+ Додати товар

ID товару	Назва товару	Ціна	Редагувати	Видалити
1	ВАЗ 2109	20000		✘
2	ВАЗ 2111	26000		✘
3	ВАЗ 2109	15000		✘
4	Alfa Romeo Giulietta	300000		✘
5	Audi A6 3.0 2011	25000		✘
6	Mitsubishi Lancer X 2008	7700		✘
7	Toyota Camry 2012	17600		✘
8	Mitsubishi Pajero Sport 2008	11200		✘
9	BMW M5 2012	25000		✘
10	BMW X6M 2017	300000		✘
11	Ford Focus 2009	17600		✘

Рисунок 3.21 – Сторінка адміністрування автомобілів

Сторінка адміністрування автомобілів містить в собі таблицю, за допомогою якої адміністратор має можливість подивитися які автомобілі зараз є в системі, їх ціну, назву й ID товару. Для того щоб додати новий автомобіль на сторінку продажу, потрібно вибрати пункт меню «Додати товар», після чого відкриється нова сторінка з формою для заповнення інформації про додаваний транспорт, яка містить в собі:

- «Назва товару»;
- «Вартість»;
- «Категорія»;
- «Виробник»;
- «Пробіг»;
- «Кузов»;
- «Інформація»;
- «Двигун»;
- «Привід»;
- «Колір»;
- «Витрати палива»;
- «Адреса продавця»;
- «Номер телефону»;
- «Зображення товару».

Сторінки додавання нового автомобіля й новий автомобіль в системі показано на рис. 3.22 – 3.23.

Адмін панель Керування автомобілями Редагувати товари
Додати новий товар

Назва товару	<input type="text" value="Mercedes-Benz GL 350"/>
Вартість, \$	<input type="text" value="42000"/>
Категорія	<input type="text" value="Mercedes"/>
Виробник	<input type="text" value="Mercedes"/>
Пробіг	<input type="text" value="120000"/>
Кузов	<input type="text" value="Седан"/>
Інформація	<input type="text" value="Автомобіль у відмінному стані"/>

Рисунок 3.22 – Сторінка додавання нового автомобіля



Рисунок 3.33 - Доданий в систему автомобіль

Також до таблиці з автомобілями додано можливість видалити або виправити характеристики вже існуючих в системі автомобілів.

Для того щоб додавати нові товари в систему за допомогою панелі адміністрування, було створено два файли «Product.php» з функцією «createProduct», та файл «AdminProductController.php» з функцією «actionCreate».

Після того як адміністратор заповнить усі поля форми й натисне кнопку «Додати товар», уся інформація потрапить в функцію «actionCreate», яка подивиться на правильність заповнення усіх полів форми додавання автомобіля і якщо помилок немає, то відправить данні у функцію «createProduct», яка за допомогою PHP та SQL запитів додасть автомобіль до таблиці бази даних.

Реалізація функцій «createProduct» та «actionCreate» показана на рис. 3.34 – 3.35.

```
// Получаем список категорий для выпадающего списка
$categoriesList = Category::getCategoriesListAdmin();

// Обработка формы
if (isset($_POST['submit'])) {
    // Если форма отправлена
    // Получаем данные из формы
    $options['name'] = $_POST['name'];
    $options['price'] = $_POST['price'];
    $options['category_id'] = $_POST['category_id'];
    $options['brand'] = $_POST['brand'];
    $options['Mileage'] = $_POST['Mileage'];
    $options['body'] = $_POST['body'];
    $options['info'] = $_POST['info'];
    $options['engine'] = $_POST['engine'];
    $options['drive_unit'] = $_POST['drive_unit'];
    $options['color'] = $_POST['color'];
    $options['fuel_consumption'] = $_POST['fuel_consumption'];
    $options['address'] = $_POST['address'];
    $options['phone_number'] = $_POST['phone_number'];
    // флаг ошибок в форме
    $errors = false;

    // При необходимости можно валидировать значения нужным образом
    if (!isset($options['name']) || empty($options['name'])) {
        $errors[] = 'Заполните поля';
    }

    if ($errors == false) {
        // Если ошибок нет
        // Добавляем новый товар
        $id = Product::createProduct($options);

        // Если запись добавлена
        if ($id) {
            // Проверим, загрузилось ли через форму изображение
            if (is_uploaded_file($_FILES['image']['tmp_name'])) {
                // Если загрузилось, переместим его в нужную папку, дадим новое имя
                move_uploaded_file($_FILES['image']['tmp_name'], $_SERVER['DOCUMENT_ROOT'] . "/upload/images/products/{$id}.jpg");
            }
        }

        // Перенаправляем пользователя на страницу управления товарами
        header("Location: /admin/product");
    }
}
```

Рисунок 3.34 – Функція «actionCreate»

```
public static function createProduct($options)
{
    // Соединение с БД
    $db = Db::getConnection();

    // Текст запроса к БД
    $sql = 'INSERT INTO product' .
        ' (' . 'name, price, category_id, brand, Mileage,' .
        ' ' . 'body, info, engine, drive_unit, color, fuel_consumption, address, phone_number)' .
        ' (' . 'VALUES ' .
        ' ' . '(:name, :price, :category_id, :brand, :Mileage, :body,' .
        ' ' . ':info, :engine, :drive_unit, :color, :fuel_consumption, :address, :phone_number)';

    // Получение и возврат результатов. Используется подготовленный запрос
    $result = $db->prepare($sql);
    $result->bindParam(':name', $options['name'], PDO::PARAM_STR);
    $result->bindParam(':price', $options['price'], PDO::PARAM_STR);
    $result->bindParam(':category_id', $options['category_id'], PDO::PARAM_STR);
    $result->bindParam(':brand', $options['brand'], PDO::PARAM_INT);
    $result->bindParam(':Mileage', $options['Mileage'], PDO::PARAM_STR);
    $result->bindParam(':body', $options['body'], PDO::PARAM_INT);
    $result->bindParam(':info', $options['info'], PDO::PARAM_STR);
    $result->bindParam(':engine', $options['engine'], PDO::PARAM_INT);
    $result->bindParam(':drive_unit', $options['drive_unit'], PDO::PARAM_INT);
    $result->bindParam(':color', $options['color'], PDO::PARAM_INT);
    $result->bindParam(':fuel_consumption', $options['fuel_consumption'], PDO::PARAM_INT);
    $result->bindParam(':address', $options['address'], PDO::PARAM_INT);
    $result->bindParam(':phone_number', $options['phone_number'], PDO::PARAM_INT);
    if ($result->execute()) {
        // Если запрос выполнен успешно, возвращаем id добавленной записи
        return $db->lastInsertId();
    }
    // Иначе возвращаем 0
    return 0;
}
```

Рисунок 3.35 – Функція «createProduct»

3.8 Висновки до розділу 3

Завданням 3 розділу було практична реалізація системи обліку і продажу транспортних засобів. Реалізація з застосуванням технологій й методів проаналізованих в 2 розділі.

У процесі створення структури, концентрувалася увага на створенні максимально зручною й привабливою системи для користувачів. Було вибрано адаптивний дизайн, як найбільш привабливий для створення інформаційно-управляючих систем в 2018 році. Адаптивну верстку було перевірено за допомогою сервісу «Mobile-Friendly-Test». Для розробки адаптивного дизайну, використовувався HTML5, CSS3 та фреймворк Bootstrap5, який прискорив і спростив швидкість розробки макетів сторінок системи обліку і продажу транспортних засобів й зробив систему максимально адаптивною й зручною для використання користувачем. Для того щоб система була максимально привабливою для користувачів, до дизайну було додано можливість зміни мови на українську й англійську.

Для серверної частини використовується динамічне відображення інформації. Так як сторінок з автомобілями може бути досить багато і вони будуть постійно додаватимуться, потрібно щоб система сама генерувала на сервері нові сторінки без втручання розробника. Для цього створено шаблон, за допомогою якого подальше будуть створюватися нові сторінки в системі, інформація яка буде автоматично генеруватися і заповнювати сторінки системи, буде зберігатися в базі даних MySQL.

Для того щоб користувачу, було зручно користуватися системою й шукати автомобілі, було створено окрему сторінку “Автомобілі”. Де є можливість переглянути усі пропозиції автомобілів на продаж, або вибрати окрему марку автомобіля і переглянути тільки ті пропозиції авто які потрібні покупцеві. Якщо користувачу буде потрібно знайти автомобіль якимось відповідним параметрам, то в системі реалізований пошук по параметрам: «Тип кузова», «Ціна», «Тип палива». Це значно полегшить користування системою, не потрібно постійно шукати автомобілі з списку, достатньо задати потрібні параметри в пошуку. Для того щоб користувач мав можливість подивитися уся інформацію про цікавий автомобіль, є можливість відкрити розширену інформацію про кожне авто.

Для управління системою, було додано сторінку “Адміністрування”, вона дасть керуючому можливість редагувати усі сторінки без входження в бази даних, все буде здійснюватися за допомогою адмін-панелі.

Це не тільки зручніше с точки зору керування, а й надійніше, адміністратору не потрібно заходити в phpMyAdmin й шукати потрібну таблицю, або на сам сервер й шукати потрібний файл, уся потрібна інформація буде показана в адмін-панелі, через яку є можливість додати, видалити, змінити інформацію.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи було "Аналіз та особливості побудови комп'ютерної системи інтернет-магазину з продажу транспортних засобів", і як результат було створено інформаційно-комп'ютерну систему "AutoShop". За цим в подальшому розроблятиметься реальна система, яка значно полегшить процес розробки комерційної інформаційно-управляючої системи з продажу автомобілів. Так як в процесі проектування використовувався комп'ютер, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися розроблена інформаційно-управляюча система.

4.1 Загальні питання з охорони праці

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. Умови праці на робочому місці, безпека машин, механізмів, стан засобів захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

4.2 Аналіз стану умов праці

Робота над створенням інформаційно-управляючої системи проходитиме в приміщенні квартири. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	6
Ширина, м	4
Висота, м	3
Площа, м ²	24
Об'єм, м ³	75

Згідно з ДСН 3.3.6.042-99 [18] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має три вікна та система загального рівномірного освітлення, що встановлена на стелі.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [19] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500

Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15°. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою люстри, забезпечує рівень освітленості не менше 200 Лк.

У кімнаті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

4.2.3 Навантаження та напруженість процесу праці

Як приклад наведено опис процесу праці оформлення роботи під час виконання магістерської роботи: за фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору,

м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявне психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- статичного;
- динамічного;

б) нервово-психічного перевантаження:

- розумового перенапруження;
- монотонності праці;
- перенапруження аналізаторів;
- емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи на ведені в ДСанПіН 3.3.2.007-98[19]

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при експлуатації, пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при експлуатації

Основними робочими характеристиками персонального комп'ютера є наступні:

- робоча напруга $U = +220\text{В} \pm 5\%$;
- робочий струм $I = 2\text{А}$;
- споживана потужність $P = 350\text{Вт}$.

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 ДСанПіН 3.3.2-007-98 [19].

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3).

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількіс на оцінка	Нормативні документи
1	2	3	A
фізичні			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[18]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[20] [21]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[17]
- недостатне освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[17]
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[16] [19]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу.	2	[16] [19]

4.3.2 Пожежна безпека

Небезпека розвитку пожежі в квартирі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, кондиціонування. Небезпека загоряння пов'язана з особливістю комп'ютерів - із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів.

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі,
- 2) системою протипожежного захисту,
- 3) організаційно-технічними заходами.

Потенційними джерелами запалювання можуть бути:

- 1) іскри і дуги короткого замикання;
- 2) електрична іскра при замиканні і розмиканні ланцюгів;
- 3) перегіви від тривалого перевантаження,
- 4) відкритий вогонь і продукти горіння,
- 5) наявність речовин, нагрітих вище за температуру самозаймання,
- 6) розрядна статична електрика.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування.

4.3.3 Електробезпека

На квартирі виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Нульовий захисний провідник використовується для заземлення електроприймачів.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості

переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [18] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони

Період року	Категорія робіт	Температура С ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

4.4.2 Освітлення

Робота на ПЕОМ може здійснюватися за таких видах освітлення:

- загальному штучному освітленні, коли відео монітори розташовуються по периметру приміщення або при центральному розташуванні робочих місць у два ряди по довжині кімнати з екранами, звернені в протилежні сторони;

- суміщене освітлення (природне + штучне) тільки при одному і трьох рядном розташуванні робочих місць, коли екран і поверхню робочого столу знаходяться перпендикулярно світла несучій стіні.

Природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 - 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цій повинен бути зліва.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IVв розряду точності (середня точність). При цьому нормована освітленість на робочому місці (Ен) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2015 [17]. Джерелом природного освітлення є сонячне світло.

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2 + b^2} \cdot S_b = (1/8 \div 1/10) \cdot S_n \quad (4.1)$$

де S_b – площа віконних прорізів, м²;

S_n – площа підлоги, м².

$$S_n = a \cdot b = 6 \cdot 4 = 24 \text{ м}^2 \quad (4.2)$$

$$S_{\text{вік}} = 1/8 \cdot 24 = 3 \text{ м}^2 \quad (4.3)$$

Приймаємо 2 вікна площею $S = 1,6 \text{ м}^2$ кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5А00 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.4):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \quad (4.4)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м²;

$S = 24 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп);

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 24 \cdot 1,15 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2 \quad (4.5)$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою вікна та установки кондиціонера HYUNDAI ARN07HSSUA/ARU07HSSUA. Цей метод забезпечує приток потрібної кількості свіжого повітря.

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення).

Розрахунок захисного заземлення.

Загальний опір захисного заземлення визначається за формулою:

$$R_{3zn} = \frac{R_3 \cdot R_n}{R_n \cdot n \cdot \eta_3 + R_3 \cdot \eta_n} \quad (4.6)$$

де R_3 - опір заземлення, якими когут бать труби, опори, кути і т.п., Ом;

$R_{ш}$ - опір опори, яке з'єднує заземлювачі, Ом;

n - кількість заземлювачів;

η_z - коефіцієнт екранування заземлювача;

приймається в межах $0,2 \div 0,9$;

$\eta_z = 0,7$

$\eta_{ш}$ - коефіцієнт екранування сполучної стійки; приймається в межах $0,1 \div 0,7$;

$\eta_{ш} = 0,5$;

Опір заземлення визначається за формулою:

$$R_z = \frac{\rho}{2\pi \cdot l} \cdot \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right) \quad (4.7)$$

де ρ - питомий опір ґрунту, залежить від типу ґрунту, Ом·м;

для піску - $400 \div 700$ Ом·м; приймаємо $\rho = 400$ Ом·м;

l - довжина заземлювача, м; для труб - 2-3 м; $l = 3$ м;

d - діаметр заземлювача, м; для труб - 0,03-0,05 м; $d = 0,05$ м;

t - відстань від середини забитого в ґрунт заземлювача до рівня землі, м; $t = 2$ м.

$$R_z = \frac{400}{2 \cdot 3,14 \cdot 3} \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 2 + 3}{4 \cdot 2 - 3} \right) = 110, \text{ Ом} \quad (4.8)$$

Опір смуги, що з'єднує заземлювачі, визначається за формулою:

$$R_{ш} = \frac{\rho}{2\pi \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot t^3}, \quad (4.9)$$

де L - довжина смуги, що з'єднує заземлювачі (м) і приблизно дорівнює периметру будівлі: $P_{буд.} = 42 \cdot 2 + 38 \cdot 2 = 160$ м; $L = 160$ м;

b - ширина смуги, м; $b = 0,03$ м;

t_1 - глибина заземлення від рівня землі, м; $t_1 = 0,5$ м.

$$R_{ш} = \frac{400}{2 \cdot 3,14 \cdot 160} \cdot \ln \frac{2 \cdot 160^2}{0,03 \cdot 0,5} = 5,99, \text{ Ом} \quad (4.10)$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3}, \quad (4.11)$$

де 4 - допустимий загальний опір, Ом;

2 - коефіцієнт сезонності.

Визначаємо загальний опір захисного заземлення:

$$R_{ззп} = \frac{110 \cdot 5,99}{5,99 \cdot 79 \cdot 0,7 + 110 \cdot 0,5} = 1,7 \text{ Ом} \quad (4.12)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{ззп} < 4 \text{ Ом}$.

3) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і пароповітряних сумішей.

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинно накопичуватися сміття, непотрібний папір, мотлох та ін. речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі.

4.6 Охорона навколишнього природного середовища

4.6.1 Загальні дані з охорони навколишнього природного середовища

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря робочої зони.

4.6.2 Визначення впливу та заходів щодо поводження з відходами ІТ галузі

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів.

Відомості про місце утворення та місце розташування відходів:

- Відходи комунальні - зовнішній сміттєвий бак для зберігання побутових відходів;
- Макулатура паперова – відро для сміття, будинок 8 квартира 41;
- Змінні носії інформації – відро для сміття, будинок 8 квартира 41;
- Батарейки та акумулятори - зовнішній сміттєвий бак для зберігання побутових відходів;

4.7 Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри,

значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

ВИСНОВКИ

Основною метою даної роботи було дослідження сучасних засобів створення систем обліку і продажу транспортних засобів, та практична реалізація такої системи.

Для досягнення мети дипломного проекту, були сформульовані і вирішені такі завдання:

- проаналізовані методи та структура побудови інформаційно-управляючих систем;
- описані цілі, що стоять перед інформаційно-управляючими системами;
- проаналізовано сучасні статті з тематики дослідження;
- описані цілі, функції, види SEO-оптимізації;
- описані різновиди серверів, основні характеристики серверів;
- проаналізовані сучасні зміни в алгоритмах пошукової оптимізації;
- визначені характеристики мовного і технічного аспектів розробки комп'ютерної системи інтернет-магазину;
- практично реалізовано комп'ютерну систему за допомогою HTML5, CSS3, Bootstrap, PHP, MySQL та патерну Model-View-Controller.

В ході виконання роботи, було проаналізовано і виконано наступні висновки:

1. Сформульовані цілі, що стоять перед інформаційно-управляючими системами для організації:

- Взаємодія з потенціальним споживачем.
- Залучення нових покупців і партнерів.
- Постійний доступ до новин фірми.
- Організація товарів.
- Мінімальні затрати.

2. Виявлені основні особливості побудови системи, які потрібно використовувати під час розробки системи:

- дизайн – це перший параметр, який оцінюють відвідувачі, тому він є одним з основних елементів;
- структура - повинна бути зручною, щоб кожен користувач зміг швидко знайти потрібну йому інформацію, гарний але незручний сайт буде приносити менше користі;
- контент – при відсутності цінної інформації ресурс не буде цікавий, але інформація повинна бути унікальною, цінною для користувачів і складатися з урахуванням вимог SEO;
- грамотність – структуровані тексти, інформативні підзаголовки, відсутність орфографічних, граматичних і стилістичних помилок не тільки додають проекту серйозність в очах користувачів, але і сильно впливають на просування сайту, так як враховуються пошуковими системами;

- швидкість завантаження - сторінки повинні відкриватися швидко, так як не всі користувачі готові миритися з повільним завантаженням сторінок, крім того, це негативно впливає на просування, швидкість залежить від професійності на етапі верстки та програмування інформаційно-управляючої системи;

- просування – дозволяє збільшити кількість відвідувачів шляхом реєстрації сайту в каталогах, розміщення контенту, посилання і багато чого іншого, врахувати всі нюанси цього процесу може тільки професіонал в області SEO.

3. Виявлено чотири основних способи створення системи, до яких відносяться, створення системи за допомогою мов програмування, конструктори, фреймворки та за допомогою CMS.

Аналізуючи основні підходи до розробки інформаційно-управляючої системи, було вибрано засіб створення системи за допомогою мов програмування, так як цей спосіб найбільш привабливим у сфері веб-програмування. Завдяки цьому способу, можна досягти максимально привабливого результату для користувача та пошукових роботів.

4. Для того щоб зрозуміти як розробляється інформаційно-управляюча система, виявлено й розібрано із яких частин складається кожен веб-сайт:

- Графічний інтерфейс - це система графічних зображень, за допомогою яких користувач може "спілкуватися" з комп'ютером. Це вікна, іконки, панелі управління - зображення, що дозволяють дати комп'ютеру команду за допомогою кліка. Для порівняння, протилежністю графічного інтерфейсу є текстовий інтерфейс, для роботи в якому користувач вводить спеціально сформульовані команди в командний рядок. Вся інформація при цьому постає в текстовому вигляді, наприклад, як при налаштуванні комп'ютера в BIOS.

- Програмна частина - це сукупність програм, які забезпечують "взаєморозуміння" між комп'ютером і користувачем. Вони обробляють запити, що надходять, готують графічний інтерфейс і видають відповідь. Наприклад, якщо користувач хоче перейти по посиланню на іншу частину сайту, саме програмна частина аналізує суть запиту і відкриває нову сторінку.

- Серверне обладнання - це комп'ютерне обладнання, оснащене спеціальним програмним забезпеченням для зберігання самої системи і даних користувачів, забезпечення безперервної доступності сайту в інтернеті і взаємодії зі сторонніми веб-службами.

5. Було проаналізовано основні різновиди розробки клієнтської частини системи, де яких відноситься фіксована верстка або статична, гумова верстка, таблична верстка або верстка таблицями, блокова, адаптивна верстка або мобільна верстка, гнучка верстка або flex верстка, семантична верстка, кросбраузерна верстка.

Для системи було вибрано адаптивну, кросбраузерну верстку з використанням flex-box та bootstrap сітки. Було проаналізовано і порівняно мобільну версію системи, та адаптивну

сітку, й вирішено використовувати адаптивну верстку, вона має швидку швидкість завантаження, не вимагає перенаправлень на ту чи іншу версію та зручна в розробці.

6. Було проаналізовано розробку динамічних систем. Й зроблено такі висновки, на динамічному веб-сайті HTML сторінки створюються отриманням даних з бази даних в плейсхолдери в HTML шаблонах, це набагато більш ефективний шлях зберігання великої кількості контенту, ніж використання статичних систем. Динамічна система може повертати різні дані в залежності від інформації, заснованої на URL, що поставляється користувачем або збережені переваги і може виробляти інші операції як частину відповіді.

7. Для об'єднання користувацької частини та серверної, було вибрано патерн Model-View-Controller. Патерн Model-View-Controller, являє собою шаблон проектування архітектури програмного забезпечення, основним завданням якого є відділення функцій роботи з даними від їх подання. Теоретично, грамотно спроектований MVC-додаток дозволить фронтенд і бекенд розробникам в ході роботи не втручатися в зони відповідальності один одного.

В результаті отриманої інформації в 1 та 2 розділі, було практично реалізовано систему обліку і продажу транспортних засобів. У процесі створення структури, концентрувалася увага на створенні максимально зручною й привабливою системи для користувачів. Адаптивну верстку було перевірено за допомогою сервісу «Mobile-Friendly-Test». Для розробки адаптивного дизайну, використовувався HTML5, CSS3 та фреймворк Bootstrap5, який прискорив і спростив швидкість розробки макетів сторінок системи обліку і продажу транспортних засобів й зробив систему максимально адаптивною й зручною для використання користувачем. Для того щоб система була максимально привабливою для користувачів, до дизайну було додано можливість зміни мови на українську й англійську. Для серверної частини використовується динамічне відображення інформації. Так як сторінок з автомобілями може бути досить багато і вони будуть постійно додаватимуться, потрібно щоб система сама генерувала на сервері нові сторінки без втручання розробника. Для цього створено шаблон, за допомогою якого подальше будуть створюватися нові сторінки в системі, інформація яка буде автоматично генеруватися і заповнювати сторінки системи, буде зберігатися в базі даних MySQL.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Евгений-Ющук – «Интернет-розвідка. Інструкція.» Видавництво «Вершина», 2007- 304 с.
2. А. Яковлев – «Розкрутка і просування сайтів: основи, секрети, трюки». Видавництво «БХВ - Петербург», 2007-328 с.
3. В.С. Ашманов Иванов А.А. – «Просування сайту в пошукових системах» Видавництво «Вільямс», 2008-304 с.
4. Mike Moran , Bill Hunt – «Search Engine Marketing, Inc.: Driving Search Traffic to Your Company's Web Site» 2015-528 pages.
5. Amy N. Langville, Carl D. Meyer – «Google's PageRank and Beyond: The Science of Search Engine Rankings» 2012-467 pages.
6. Jennifer Grappone, Gradiva Couzin – «Search Engine Optimization: An Hour a Day» 2008-384 pages.
7. Богомоллов О.Б. – «Web-конструювання на HTML: практикум» Видавництво «БІНОМ. Лаборатория знаний», 2008-287 с.
8. Фаулер М. – «Рефакторинг: поліпшення існуючого коду.» Видавництво «Символ Плюс», 2003. – 432 с.
9. Клименко Р. А. – «Веб-мастеринг на 100%» Видавництво «Питер», 2013. — 512 с.
10. Internet World Stats. Usage and Population Statistics [Електронний ресурс] – 2018. – Режим доступу: <http://www.internetworldstats.com/stats.htm>. – Назва з екрану.
11. Етан Браун - «Вивчаємо JavaScript. Керівництво по створенню сучасних веб-сайтів» Видавництво «Вільямс», 2016. — 368 с.
12. Евдокимов П.В. - «Створення сайтів своїми руками на BOOTSTRAP» Видавництво «НіТ», 2017. — 240 с.
13. Марк Соупер, Скотт Мюллер - «Модернізація і ремонт серверів» Видавництво «Диалектика Вільямс», 2013. — 976 с.
14. Люк Веллинг, Лаура Томсон - «Розробка веб-додатків за допомогою PHP і MySQL» Видавництво «Вільямс», 2016. — 768 с.
15. Пауерс Шелли - «Вивчаємо Node. Переходимо на сторону сервера» Видавництво «Вільямс», 2016. — 579 с.
16. ДК 005-96 Державний класифікатор України. Класифікатор відходів
17. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
18. ДБН В.2.5-28:2015 Природне і штучне освітлення

19. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих
20. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин
21. ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление
22. ГОСТ 13109-97 „Электрическая энергия. Совместимость технических средств электромагнитных. Нормы качества электроэнергоснабжения общего назначения”.
23. ГОСТ 13109-97 „Электрическая энергия. Совместимость технических средств электромагнитных. Нормы качества электроэнергоснабжения общего назначения”
24. ДБН В.2.5-28:2015 Природне і штучне освітлення
25. Кветкин П.В. «Аналіз сучасного напрямку розвитку веб-технологій», Комунальна державна установа.
26. Пасюков А.А., Якимов А.С. «Порівняння продуктивності серверних мов програмування»,
Інформатизація інженерної освіти ІНФОРІНО 2014, с. 91 – 94.
27. Добровольский А.А., «Сучасні принципи клієнтоорієнтованості користувацького інтерфейсу», «Менеджмент організацій», с 197 - 199.
28. Голуб А.Н., «Електронна комерція», «Економіка і соціум» 2016, с. 56 – 58.
29. Стиров С.Е. «Аналіз та особливості побудови комп'ютерної системи інтернет-магазину з продажу транспортних засобів», Конференція: «Майбутній науковець»
30. Стиров С.Е. «Дослідження та розробка комп'ютерної системи обліку і продажу транспортних засобів », збірник: «Збірник науково практичних праць ІТ-Ідея 2018» с. 88 – 91.
31. Сучасний підручник JavaScript [Електронний ресурс] – 2018. – Режим доступу: <https://learn.javascript.ru/> – Назва з екрану.
32. Що таке Flexbox? Опис всіх css властивостей, основні принципи, переваги та недоліки. [Електронний ресурс] – 2017. – Режим доступу: <https://html5.by/blog/flexbox/> – Назва з екрану.
33. JavaScript ядро. [Електронний ресурс] – 2016. – Режим доступу: <http://dmitrysoshnikov.com/ecmascript/ru-javascript-the-core/> – Назва з екрану.
34. Exploring javascript by example [Електронний ресурс] – 2018. – Режим доступу: <http://perfectionkills.com/know-thy-reference/> – Назва з екрану.
35. Як зробити аналіз конкурентів інтернет магазину [Електронний ресурс] – 2016. – Режим доступу: <https://shopconstructor.ru/kak-sdelat-analiz-konkurentov-internet-magazina/> – Назва з екрану.

36. SEO-аналіз сайтів конкурентів - вичерпне керівництво і сервіси. [Електронний ресурс] – 2017. – Режим доступу: <https://q-seo.com.ua/blog/analiz-saytov-konkurentov/> – Назва з екрану.
37. Воронюк А., «Актуальний інтернет маркетинг», Видавництво «ІРІО», 2018. — 160 с.
38. Scot Hawkins, «Apache Web Server Administration and e-Commerce Handbook», Видавництво «Paperback», 2008. — 270 с.
39. Інформаційно-управляюча система. [Електронний ресурс] – 2015. – Режим доступу: <http://www.bibliotekar.ru/7-robot/53.htm> – Назва з екрану.
40. Адаптивна верстка: що це і як використовувати. [Електронний ресурс] – 2018. – Режим доступу: <https://tproger.ru/translations/responsive-web-design-tips/> – Назва з екрану.
41. Брайан Хоган, «HTML5 та CSS3. Веб-розробка по стандартам нового покоління», Видавництво «ПІТЕР», 2012. — 272 с.
42. Sass & Compass For Designers. [Електронний ресурс] – 2018. – Режим доступу: <http://sassandcompass.com/> – Назва з екрану.
43. Розробка динамічних систем. [Електронний ресурс] – 2014. – Режим доступу: http://citforum.ck.ua/internet/articles/dyn_web.shtml – Назва з екрану.
44. Pitt, Chris, «Pro PHP MVC», Видавництво «APRESS», 2012. — 272 с.
45. Колісніченко Д., «Движок для вашого сайту. CMS Joomla!, Slaed, PHP-Nuke», Видавництво «bhv», 2017. — 328 с.
46. Індексція сайту в пошукових системах. [Електронний ресурс] – 2014. – Режим доступу: <https://seoprofy.ua/blog/optimizaciya-sajtov/indeksaciya-sajta> – Назва з екрану.
47. Neil Grey, «Web Server Programming», Видавництво «Neil Gray», 2013. — 246 с.
48. Ken Coar, Rich Bowen, «Apache Cookbook, 2nd Edition», Видавництво «O'RELLY», 2009. — 320 с.

ДОДАТОК А

Лістинг програми

Файл db_params.php

```
<?php
return array(
    'host' => 'localhost',
    'dbname' => 'AutoShopUa',
    'user' => 'root',
    'password' => "",
);
```

Файл routes.php

```
<?php
return array(
    // Автомобілі
    'product/([0-9]+)' => 'product/index/$1',
    // Новини
    'contant/([0-9]+)' => 'contant/index/$1',
    'news' => 'news/index',
    'categoryNew/([0-9]+)/page-([0-9]+)' => 'news/category/$1/$2',
    'categoryNew/([0-9]+)' => 'news/category/$1', // actionCategory в ContantController
    //Запчастини
    'spareparts' => 'spareparts/index',
    'categoryPart/([0-9]+)/page-([0-9]+)' => 'spareparts/category/$1/$2', // actionCategory в
CatalogController
    'categoryPart/([0-9]+)' => 'spareparts/category/$1', // actionCategory в CatalogController
    //Каталог авто
    'catalog' => 'catalog/index', // actionIndex в CatalogController
    'category/([0-9]+)/page-([0-9]+)' => 'catalog/category/$1/$2', // actionCategory в CatalogController
    'category/([0-9]+)' => 'catalog/category/$1', // actionCategory в CatalogController
    //Реєстрація - авторизація
    'user/register' => 'user/register',
```

```
'user/login' => 'user/login',
```

Продовження додатку А

Файл AdminBase.php

```
<?php
abstract class AdminBase
{
    public static function checkAdmin()
    {
        // Проверяем авторизован ли пользователь. Если нет, он будет переадресован
        $userId = User::checkLogged();
        // Получаем информацию о текущем пользователе
        $user = User::getUserById($userId);
        // Если роль текущего пользователя "admin", пускаем его в админпанель
        if ($user['role'] == 'admin') {
            return true;
        }
        // Иначе завершаем работу с сообщением об закрытом доступе
        die('Access denied');
    }
}
```

Файл User.php

```
<?php
class User
{
    /**
     * Регистрация пользователя
     */
    public static function register($name, $email, $password)
    {
        $db = Db::getConnection();
        $sql = 'INSERT INTO user (name, email, password) '
            . 'VALUES (:name, :email, :password)';
```

Продовження додатку А

```

$result = $db->prepare($sql);
$result->bindParam(':name', $name, PDO::PARAM_STR);
$result->bindParam(':email', $email, PDO::PARAM_STR);
$result->bindParam(':password', $password, PDO::PARAM_STR);
return $result->execute();
}
/**
 * Редактирование данных пользователя
 */
public static function edit($id, $name, $password)
{
    $db = Db::getConnection();
    $sql = "UPDATE user SET name = :name, password = :password WHERE id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':password', $password, PDO::PARAM_STR);
    return $result->execute();
}

```

Файл index.php

```

<?php include ROOT.'/views/layouts/headerAdmin.php'; ?>
    <ul class="adminInfo">
        <li><a href="/admin/product"><i class="fa fa-circle"> Автомобілі</i></a></li>
        <li><a href="/admin/categoryAuto"><i class="fa fa-circle"> Категорії
автомобілів</i></a></li>
        <li><a href="/admin/contant"><i class="fa fa-circle"> Новини</i></a></li>
        <li><a href="/admin/categoryNews"><i class="fa fa-circle"> Категорії
новин</i></a></li>
        <li><a href="/admin/spareparts"><i class="fa fa-circle"> Запчастини</i></a></li>
        <li><a href="/admin/categoryParts"><i class="fa fa-circle"> Категорії
запчастин</i></a></li>
        <li><a href="/admin/order"><i class="fa fa-circle"> Замовлення</i></a></li>

```


</div>

</div>

</div>

</body>

Продовження додатку А

ДОДАТОК Б



Рисунок Б.1 – Титульний слайд

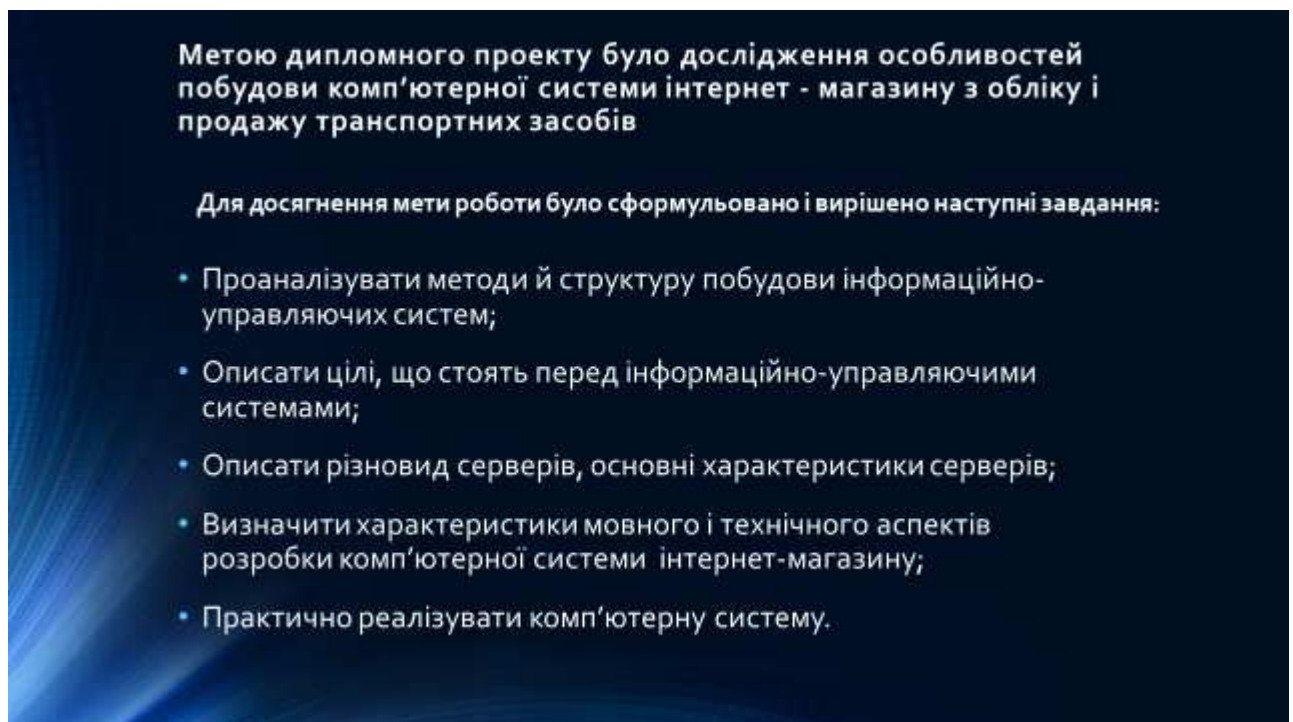


Рисунок Б.2 – Мета і завдання дослідження

Продовження додатку Б

Цілі що стоять перед інформаційно-управляючою системою:

- Взаємодія з потенціальним споживачем;
- Залучення нових покупців і партнерів;
- Новини про зміни в компанії;
- Організація товарів;
- Мінімальні затрати.

Рисунок Б.3 – Цілі що стоять перед інформаційно-управляючою системою

Основні вимоги до побудови інформаційно-управляючої системи:

- Дизайн;
- Структура;
- Контент;
- Грамотність;
- Швидкість завантаження;
- SEO просування.

Рисунок Б.4 – Основні вимоги до побудови інформаційно-управляючої системи

Продовження додатку Б

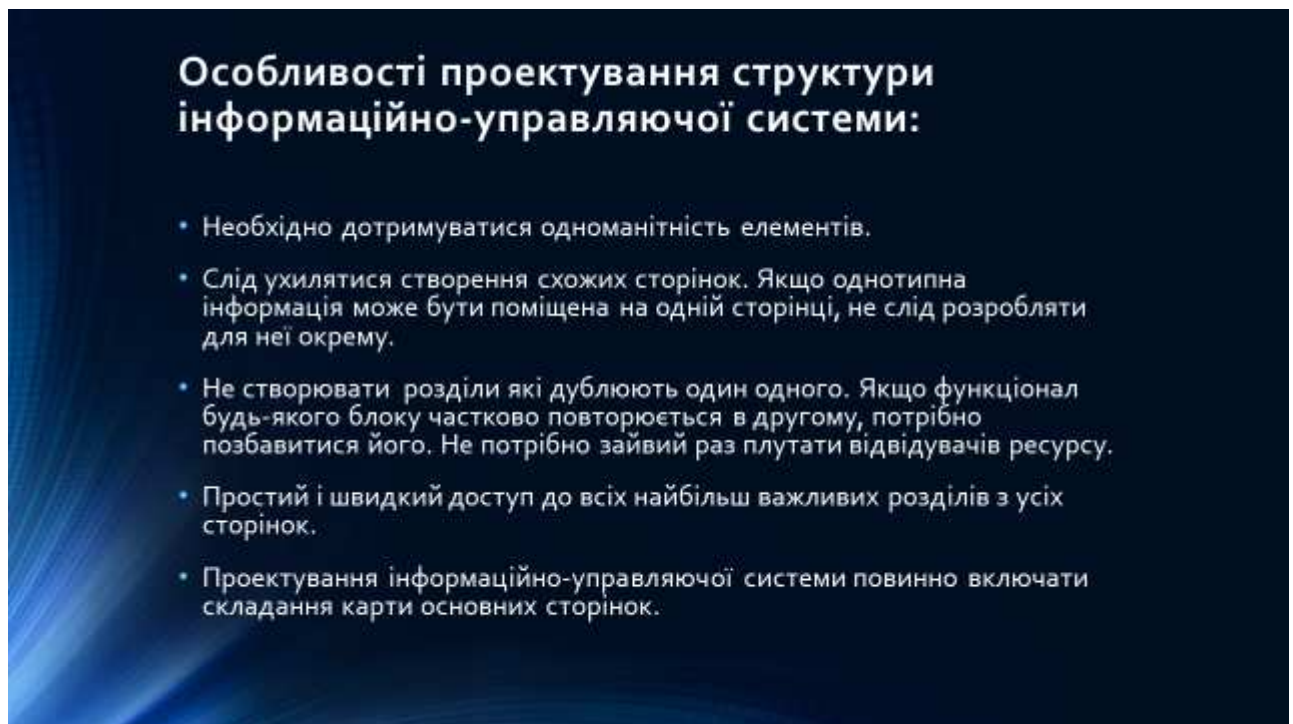


Рисунок Б.5 – Особливості проектування структури інформаційно-управляючої системи

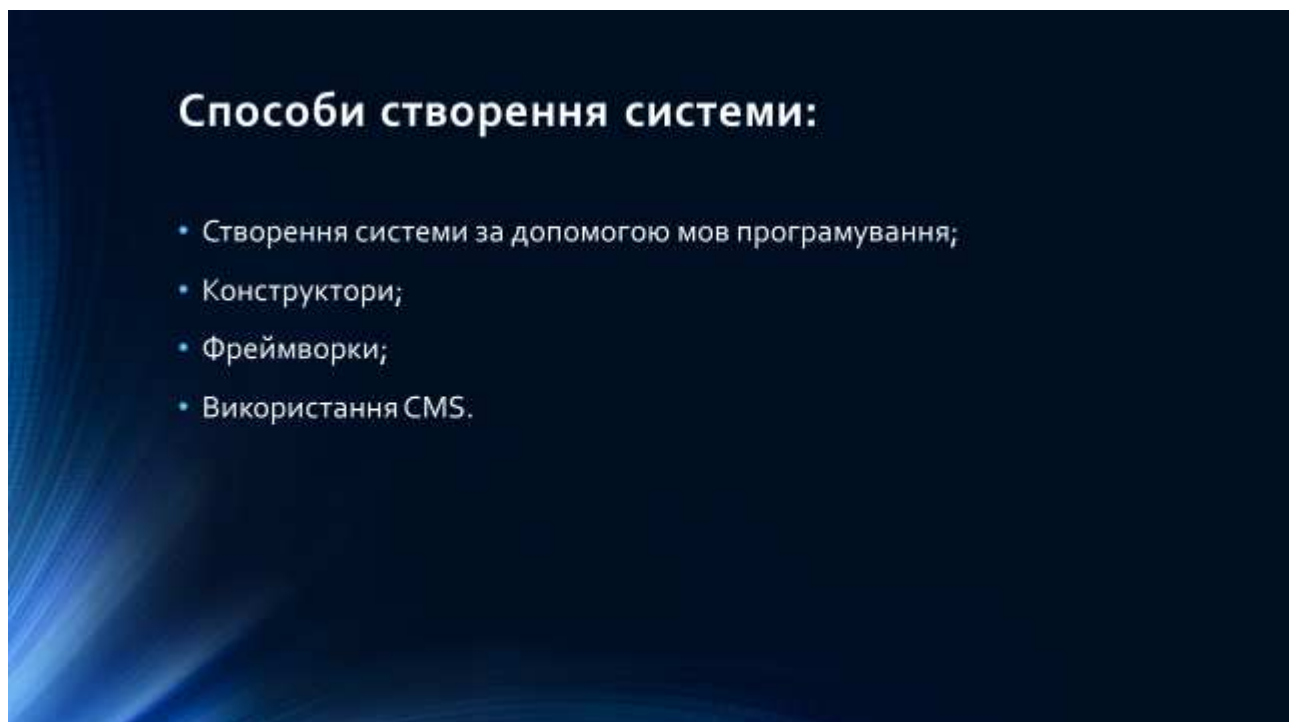


Рисунок Б.6 – Способи створення системи

Формулювання проблеми дослідження і завдань роботи

Основними проблемами при побудові системи обліку та продажу, стає, неправильно спроектована структура системи, неправильно обрані технології розробки системи, нехтування Seo оптимізацією.

Для вирішення цих проблем, було сформульовано такі основні завдання:

- Проаналізувати актуальні технології й методи розробки інформаційно-управляючих систем;
- Порівняти основні технології розробки інформаційно-управляючих систем;
- Практично реалізувати систему обліку та продажу транспортних засобів.

Рисунок Б.7 – Формулювання проблеми дослідження і завдань роботи

Різновиди верстки системи:

- Фіксована верстка або статична;
- Гумова верстка;
- Таблична верстка або верстка таблицями;
- Блокова верстка, верстка блоками або div-верстка;
- Адаптивна верстка або мобільна верстка;
- Гнучка верстка або flex верстка;
- Семантична верстка;
- Кросбраузерна верстка.



Рисунок Б.8 – різновиди верстки системи

Продовження додатку Б

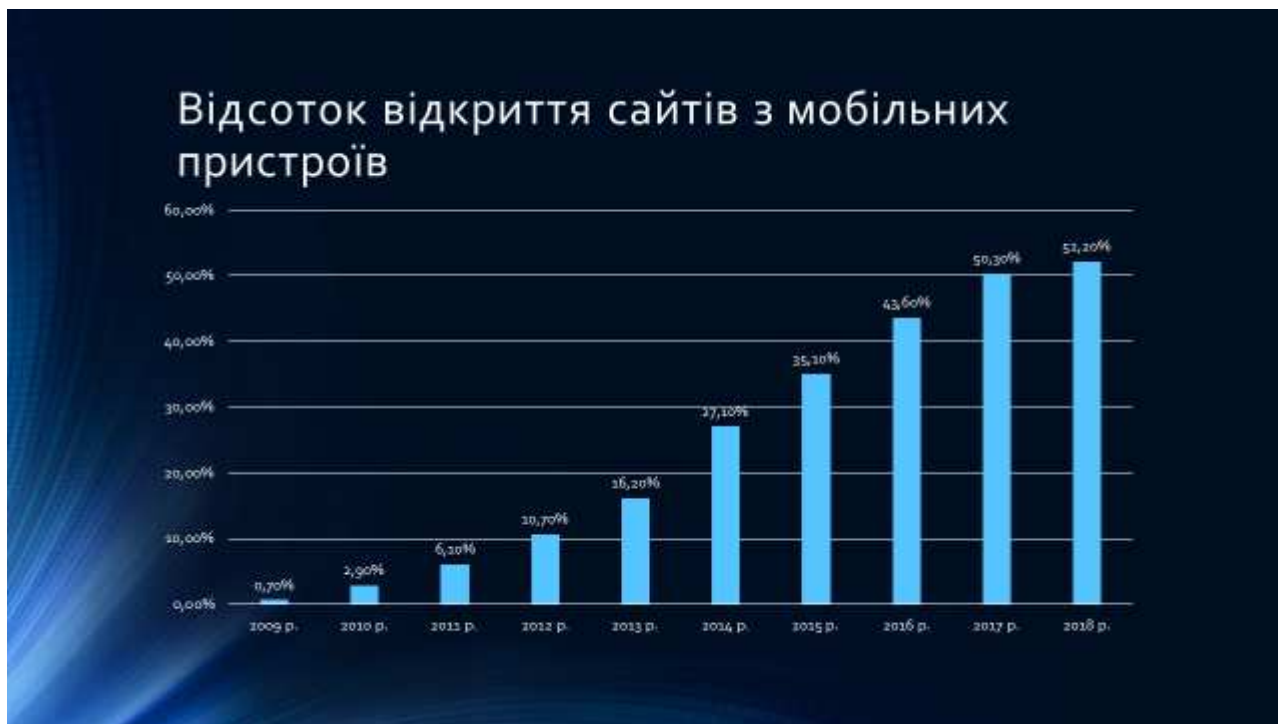


Рисунок Б.9 – Відсоток відкриття сайтів з мобільних пристроїв

Адаптивність дизайну веб-системи

- Значно поліпшити використання інтернет-системи, і підвищується трафік користувачів;
- Поліпшити розробку інтерфейсу системи, не потрібно створювати багато інтерфейсів під різні пристрої;
- Адаптувати систему при повороті екрану.

Рисунок Б.10 – Адаптивність дизайну веб-системи

Продовження додатку Б

Порівняння адаптивної сітки та мобільної версії системи

	Мобільна версія	Адаптивна сітка
Переваги	<ol style="list-style-type: none"> 1. Мале споживання інтернет трафіку. 2. Можливість окремо просувати мобільну версію сайту. 	<ol style="list-style-type: none"> 1. Швидка швидкість завантаження, не вимагає перенаправлень на ту чи іншу версію. 2. Єдина адаптуєма версія кожної сторінки. 3. Зручність розробки та наповнення сторінки.
Недоліки	<ol style="list-style-type: none"> 1. Однакові сторінки можуть негативно відбитися на просуванні сайту. 2. Збільшення навантаження на пошукових роботів. 3. SEO просування потрібно реалізовувати окремо від основної системи. 4. Мобільна версія створюється окремо під кожен гаджет, це робить її створення технічно більш складною і дорогою. 	<p>Для інформаційно насичених ресурсів буває проблематично розмістити на екрані мобільного пристрою весь вміст системи.</p>

Рисунок Б.11 – Порівняння адаптивної сітки та мобільної версії системи

Технології розробки користувацької частини системи

- Мова гіпертекстової розмітки HTML5;
- Каскадні таблиці стилів CSS3;
- Препроцесор SASS;
- Адаптивна сітка Bootstrap;
- Мова програмування JavaScript.



Рисунок Б.12 – Технології розробки користувацької частини системи

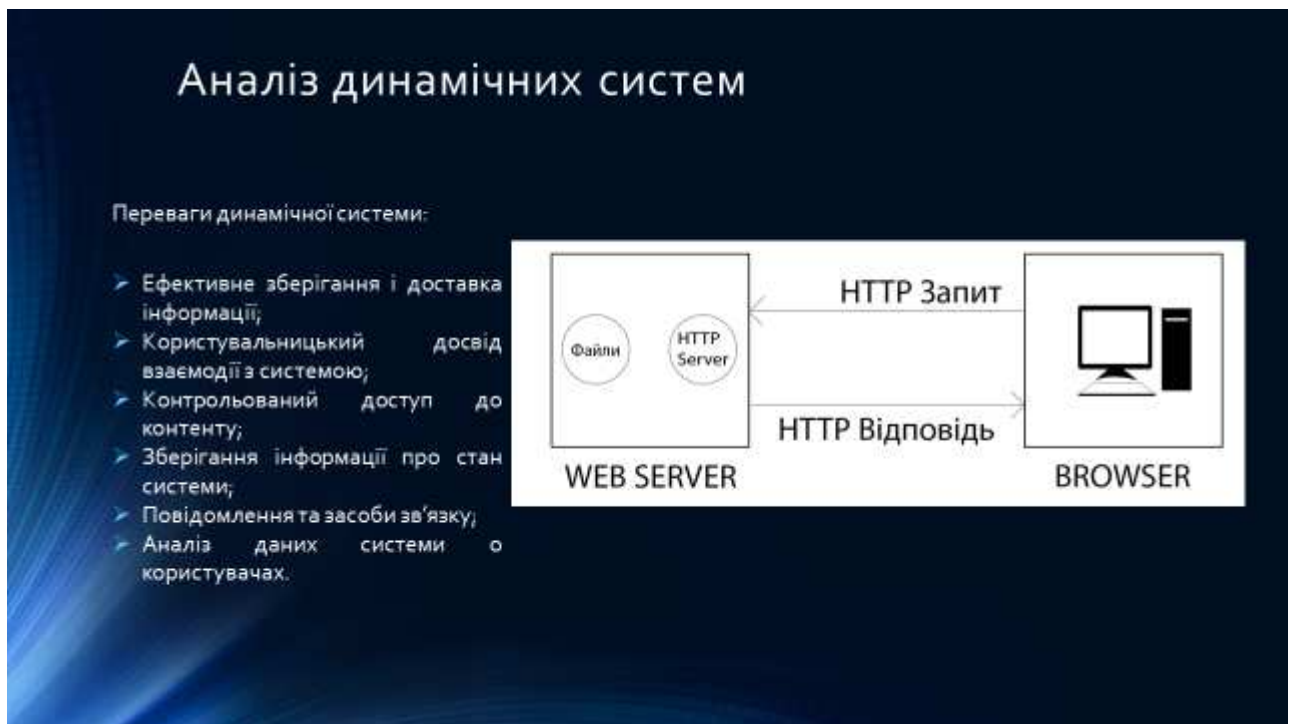


Рисунок Б.13 – Аналіз динамічних систем

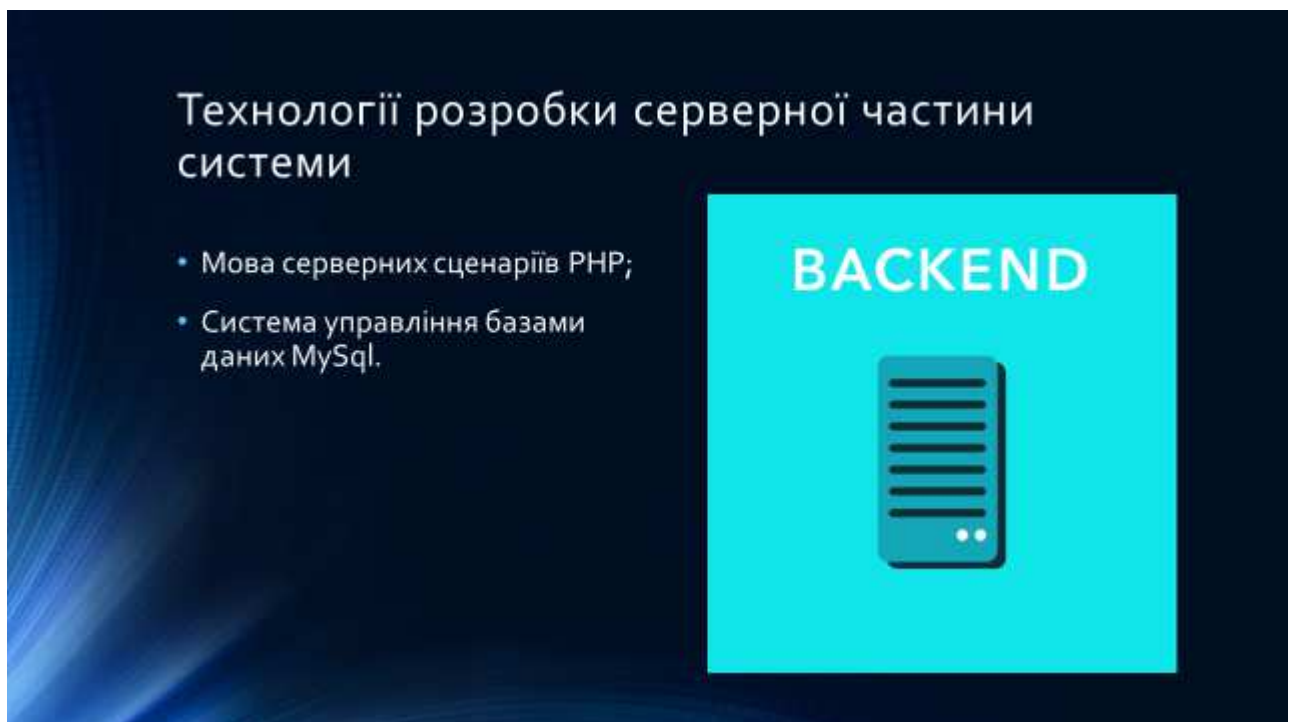


Рисунок Б.14 – Технології розробки серверної частини системи

Продовження додатку Б

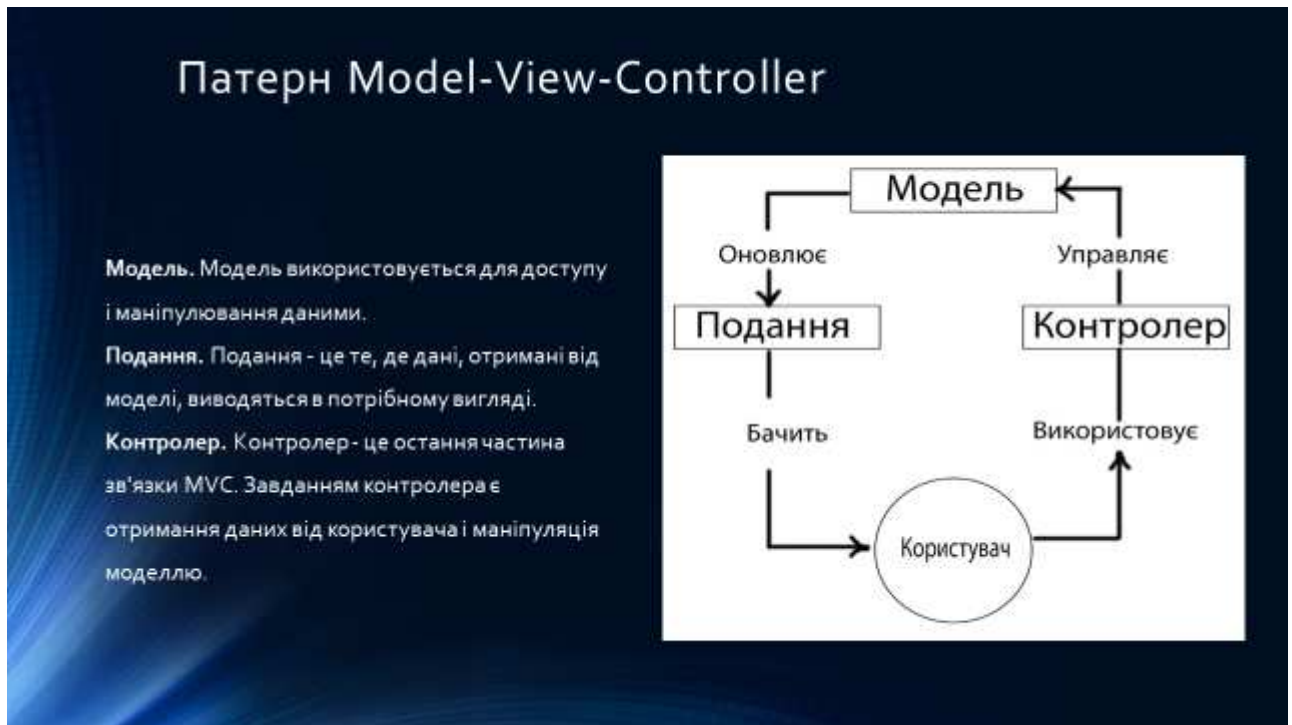


Рисунок Б.15 – Патерн Model-View-Controller

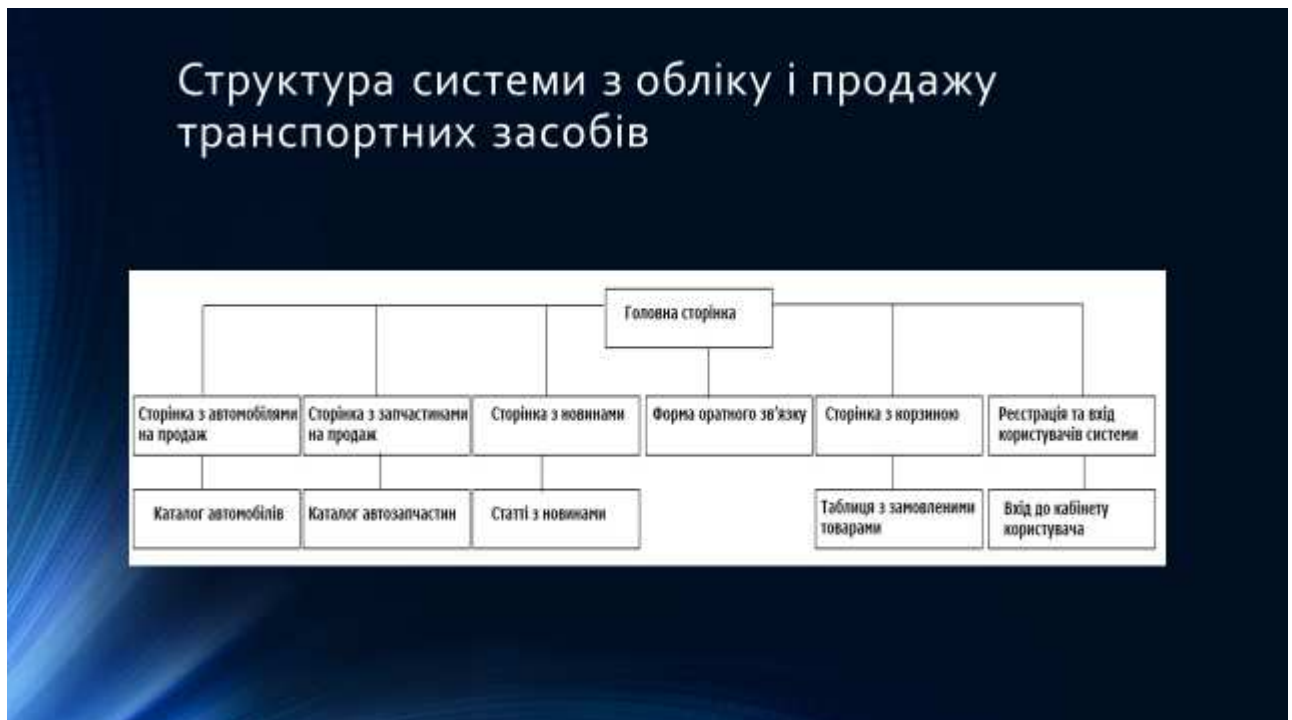


Рисунок Б.16 – Структура системи з обліку і продажу транспортних засобів

Продовження додатку Б



Рисунок Б.17 – Головна сторінка системи

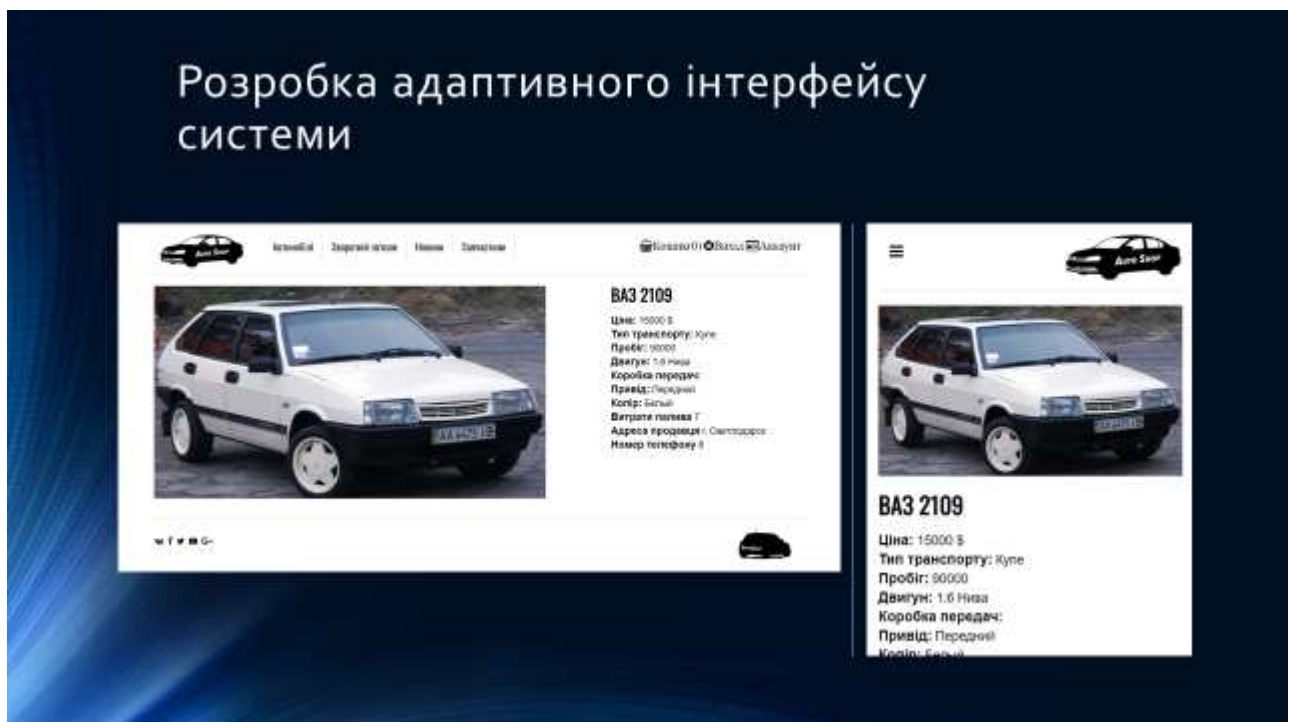


Рисунок Б.18 – Розробка адаптивного інтерфейсу системи

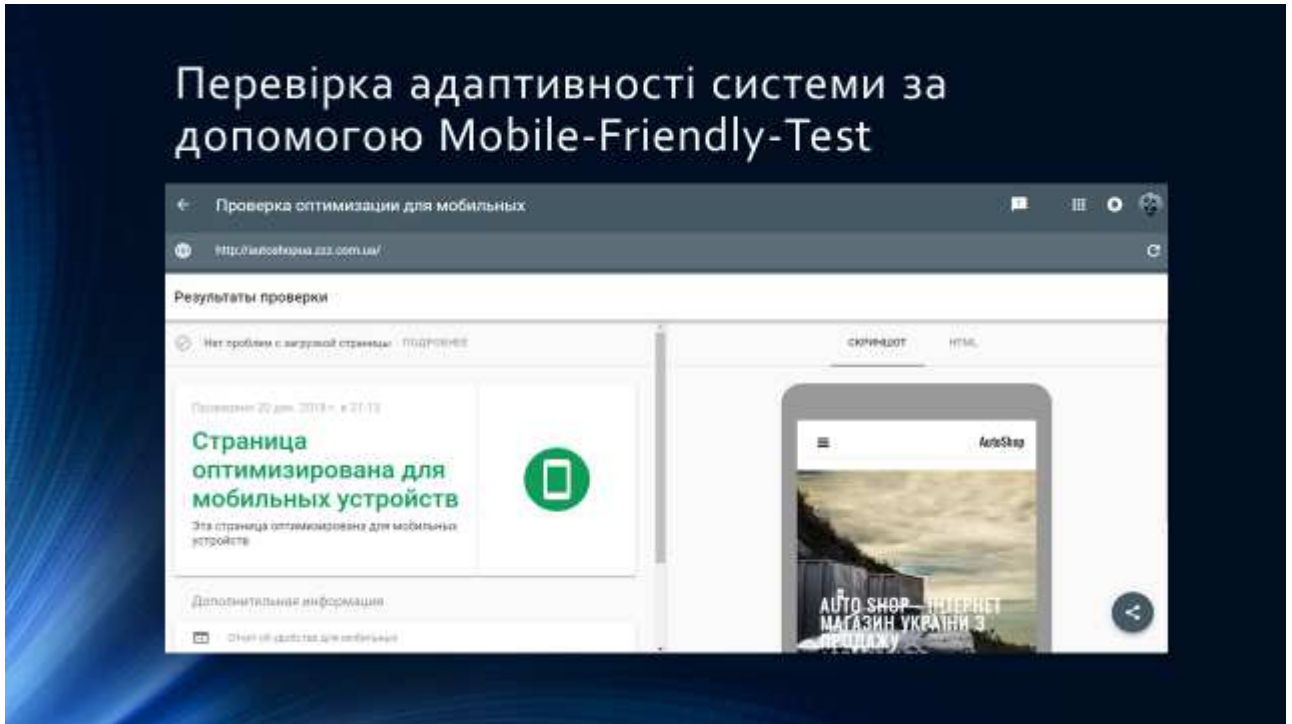


Рисунок Б.19 – Перевірка адаптивності системи за допомогою Mobile-Friendly-Test



Рисунок Б.20 – Схема реєстрації та входу в систему

Продовження додатку Б

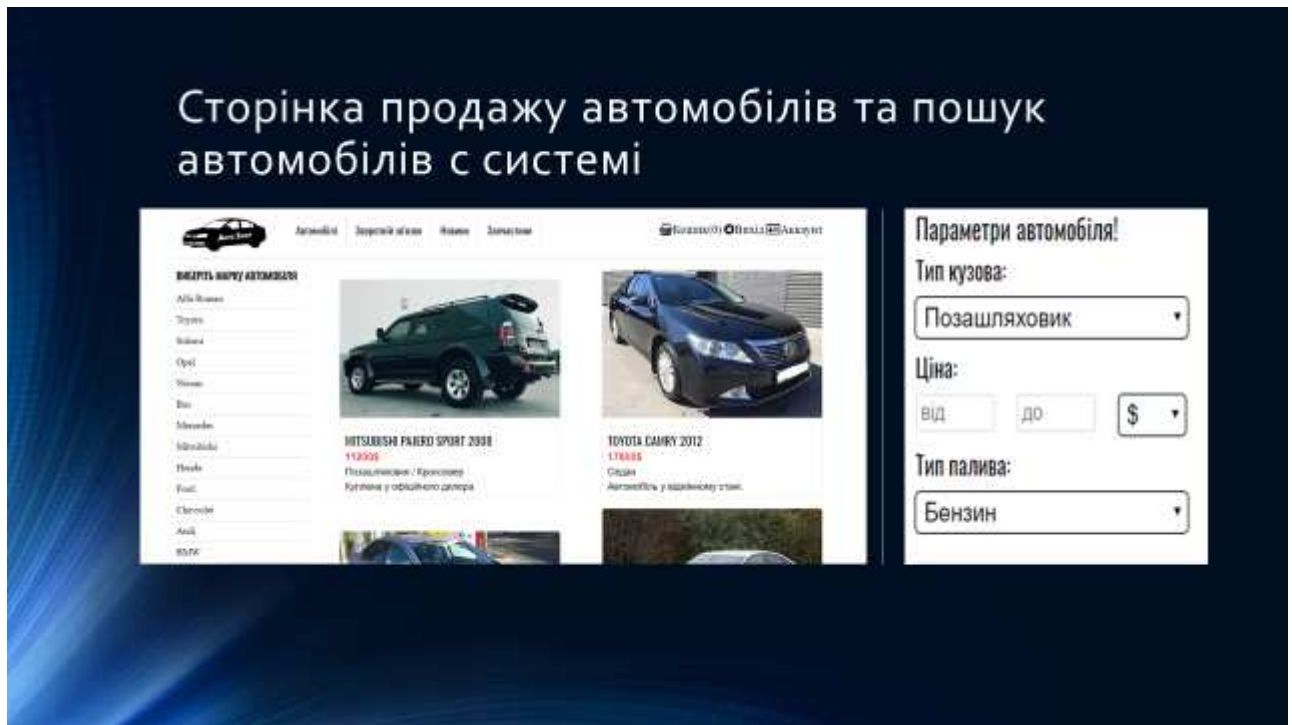


Рисунок Б.21 – Сторінка продажу автомобілів та пошук автомобілів в системі

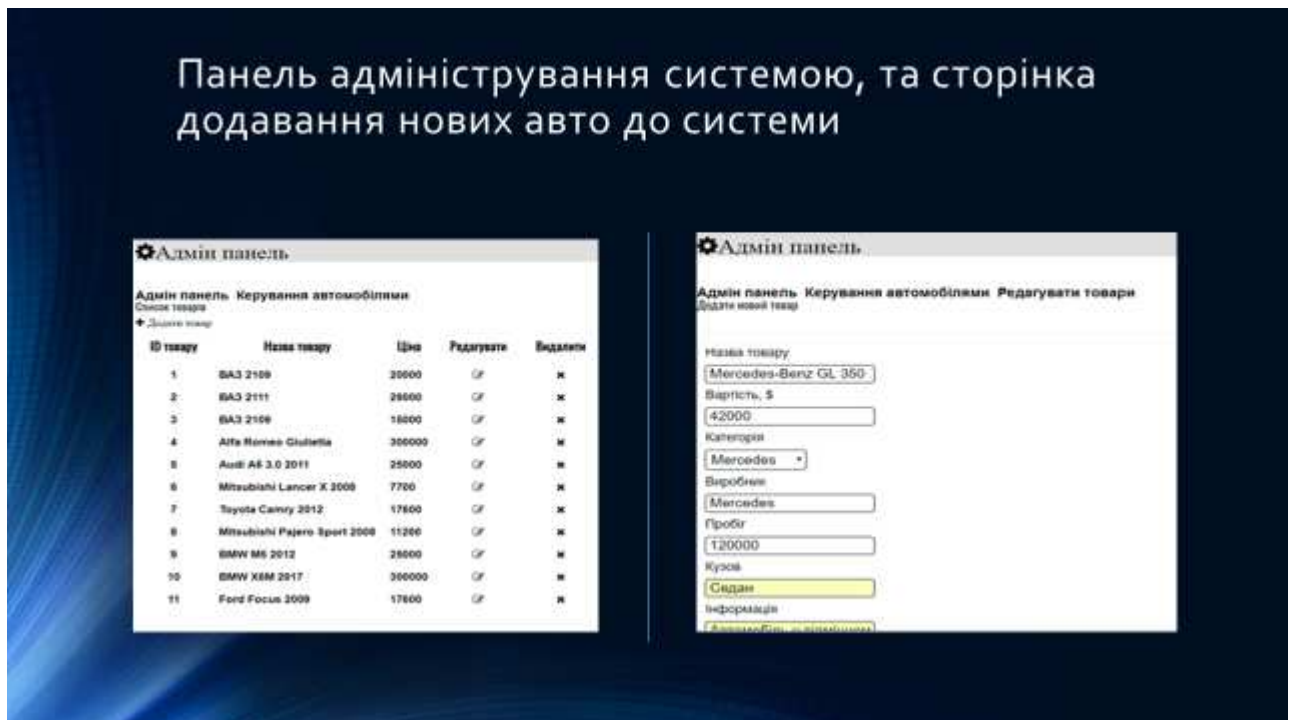


Рисунок Б.22 – Панель адміністрування системою, та сторінка додавання нових авто до системи

Висновки

Мета та завдання проекту були досягнуті та виконані в повному обсязі:

- Проаналізовані методи та структура побудови інформаційно-управляючих систем;
- Описані цілі, що стоять перед інформаційно-управляючими системами;
- Проаналізовано сучасні статті з тематики дослідження;
- Описані різновиди серверів, основні характеристики серверів;
- Визначені характеристики мовного і технічного аспектів розробки комп'ютерної системи інтернет-магазину;
- Практично реалізовано комп'ютерну систему за допомогою HTML5, CSS3, Bootstrap, PHP, MySQL та патерну Model-View-Controller.

Рисунок Б.23 - Висновки