

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сайту для сервісу з ремонту комп'ютерної
техніки «ПК Майстер»

Виконав: студент IV курсу, групи СНс-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Валицький В. М.
(підпис) (прізвище та ініціали)

Керівник Млинко Б. Б.
(підпис) (прізвище та ініціали)

Нормоконтроль Шимчук Г.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент Пастух О.А.
(підпис) (прізвище та ініціали)

Тернопіль
2021

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		Виконано
2.	Огляд джерел та аналіз за темою кваліфікаційної роботи		Виконано
3.	Виконання дослідження щодо розроблення веб-сайту для сервісу комп'ютерної техніки «ПК Майстер»		Виконано
4.	Розроблення веб-сайту для сервісу комп'ютерної техніки «ПК Майстер»		Виконано
5.	Оформлення розділу «Постановка задачі на розробку веб-сайту»		Виконано
6.	Оформлення розділу «Розробка веб-сайту для сервісу комп'ютерної техніки «ПК Майстер»»		Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»		Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»		Виконано
9.	Оформлення кваліфікаційної роботи		Виконано
10.	Нормоконтроль		Виконано
11.	Перевірка на плагіат		Виконано
12.	Попередній захист кваліфікаційної роботи		Виконано
13.	Захист кваліфікаційної роботи		

Студент

(підпис)

Валицький В. М.

(прізвище та ініціали)

Керівник роботи

(підпис)

Млинко Б. Б.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-сайту для сервісу з ремонту комп'ютерної техніки «ПК Майстер»
// Кваліфікаційна робота освітнього рівня «Бакалавр» //Валицький Василь Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2021 // 88 С., рис. – 20, табл. – 3, додат. – 8, бібліогр. – 31.

Ключові слова: база даних, гіпертекстовий процесор, веб-сервер, МРА – багатосторінковий веб-сайт, тег, скрипт, плагін

Робота присвячена розробці веб-сайту для сервісу з ремонту комп'ютерної техніки «ПК Майстер».

В першому розділі кваліфікаційної роботи наведено постановку задачі на розробку веб-сайту для сервісу комп'ютерної техніки «ПК Майстер», здійснено аналіз відомих рішень, оглянуто програмне забезпечення для даного веб-сайту, виконано постановку задачі.

В другому розділі кваліфікаційної роботи покроково описано розробку технічного та програмного продукту, наведено інструкцію щодо його обслуговування.

В третьому розділі кваліфікаційної роботи розглянуто питання з безпеки життєдіяльності, основи охорони праці: шляхи підвищення життєдіяльності людини, економічне значення заходів щодо покращенню умов та охорони праці, проведення інструктажів з охорони праці.

ANNOTATION

Website development for computer equipment repair service « PK Master »
//Qualification work of educational level « Bachelor »// Valytsky Vasyl Mykhailovych
// Ternopil National Technical University named after Ivan Pulyuy, Faculty of
Computer Information Systems and Software Engineering, Department of Computer
Science, SNC-42 Group // Ternopil, 2021 // 88 C., fig. - 20, table. - 3, appendix. -8,
bibliogr. - 31.

Keywords: database, hypertext processor, web server, MPA – multi-page
website, tag, script, plugin

The work is devoted to the development of a website for the service of repair of
computer equipment "PC Master".

The first section of the qualification work presents the task of developing a
website for the computer service "PC Master", analyzed the known solutions, reviewed
the software for this website, performed the task.

The second section of the qualification work describes step by step the
development of hardware and software product, provides instructions for its
maintenance.

The third section of the qualification work deals with issues of life safety, basics
of labor protection: ways to improve human life, the economic importance of measures
to improve working conditions and safety, conducting briefings on labor protection.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПК – персональний комп'ютер.

HTML (англ. HyperText Markup Language) – мова розмітки гіпертексту

PHP (англ. PHP: Hypertext Preprocessor) – гіпертекстовий процесор.

JS (англ. Java Script) – об'єктно-орієнтована мова програмування.

БД – база даних.

MVC (Model-View-Controller) – схема розподілених даних додатку.

ОС – (англ. operating system, OS) – операційна система.

ВВ-2 – вогнегасник вуглекислотний переносний.

ВП-2 – вогнегасник порошковий.

URL (англ. Uniform Resource Locator) – єдиний вказівник на ресурс, URL

MPA (Multiple Page Application) – багатосторінковий веб-сайт.

SPA (Single page Application) – односторінковий веб-сайт.

FTP (File Transfer Protocol, українською протокол передавання файлів).

SSH (Secure SHell – «безпечна оболонка»).

CSS (Cascading Style Sheet) – таблиця класичних стилів.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБ-САЙТУ ДЛЯ СЕРВІСУ З РЕМОНТУ КОМП'ЮТЕРНОЇ ТЕХНІКИ «ПК МАЙСТЕР».....	9
1.1 Аналіз відомих рішень.....	9
1.2 Огляд програмного забезпечення та мов програмуванн.....	18
1.2.1 Програмне забезпечення MySQL та сервер Apache.....	18
1.2.2 Мова програмування PHP.....	19
1.2.3 Мова програмування JavaScript.....	21
1.2.4 Мова розмітки гіпертексту HTML.....	22
1.2.5 Класичні таблиці стилів CSS.....	23
1.3 Постановка задачі.....	24
1.4 Висновок до першого розділу.....	25
РОЗДІЛ 2. РОЗРОБКА ВЕБ-САЙТУ ДЛЯ СЕРВІСУ З РЕМОНТУ КОМП'ЮТЕРНОЇ ТЕХНІКИ «ПК МАЙСТЕР».....	26
2.1 Розробка структури веб-сайту та структури бази даних.....	26
2.2 Програмування веб-сайту.....	34
2.3 Тестування веб-сайту та інструкція щодо його обслуговування.....	37
2.4 Висновки до другого розділу.....	50
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ...51	
3.1 Шляхи підвищення життєдіяльності людини	52
3.2 Економічне значення заходів щодо покращенню умов та охорони праці.....	53
3.3 Проведення інструктажів з охорони праці.....	56
3.4 Висновок до третього розділу.....	59
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ.....	61
ДОДАТКИ	

ВСТУП

Сучасні технології дозволяють раціонально організувати бізнес в будь-якій галузі, дистанційно керувати різними робочими процесами та виконувати інші завдання з мінімальними витратами праці.

Інтернет забезпечує величезні переваги для магазинів та підприємців, які створюють персональні веб-сайти, тим самим збільшуючи кількість потенційних клієнтів та партнерів, та заощаджуючи кошти на оренду торгових приміщень і витрати на рекламу.

Досить часто в останні роки чуємо слово «веб-сайт». Також мережа «Інтернет» пропонує значні переваги і для самих магазинів та підприємців, які створюють персональні веб-сайти, збільшуючи тим самим кількість потенційних клієнтів та партнерів, а також економлячи кошти на оренду торгових приміщень та рекламі.

Розробка багатосторінкового веб-сайту для сервісу з ремонту комп'ютерної техніки «ПК Майстер» є актуальною на сьогодні для розроблення, за рахунок того, що сучасні технології дозволяють раціонально організувати бізнес в будь-якій галузі, дистанційно керувати різними робочими процесами та виконувати інші завдання з мінімальними витратами праці. Інтернет забезпечує величезні переваги для магазинів та підприємців, які створюють персональні веб-сайти, тим самим збільшуючи кількість потенційних клієнтів та партнерів, та заощаджуючи кошти на оренду торгових приміщень та витрати на рекламу.

Найбільш практичним застосуванням у сучасному світі є веб-сайт, оскільки майже кожна компанія вважає за доцільне мати власний веб-сайт, він підкреслить успіх компанії, тобто її обличчя. Сучасні люди не мають багато часу на пошук послуг чи магазинів. Тому можливість замовляти товари та послуги, не виходячи з комп'ютера, значно розширює можливості замовника.

Використовувати технології інтернет можна не тільки в особистих цілях, але і для успішного ведення бізнесу. За допомогою спеціального програмного

забезпечення, персональних комп'ютерів і додаткового обладнання можна організувати повноцінний і високоефективний робочий процес.

Мета кваліфікаційної роботи: розробити веб-сайт для сервісу з ремонту комп'ютерної техніки «ПК Майстер».

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- здійснити аналіз відомих рішень;
- провести огляд програмного забезпечення;
- розробити структуру веб-сайту та структуру бази даних;
- здійснити програмування веб-сайту;
- провести тестування веб-сайту та інструкцію щодо його обслуговування.

Практична значимість роботи полягає в тому, що сайт практично впроваджено є корисним для забезпечення ефективної роботи сервісу.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБ-САЙТУ ДЛЯ СЕРВІСУ КОМП'ЮТЕРНОЇ ТЕХНІКИ «ПК МАЙСТЕР»

1.1 Аналіз відомих рішень

Сьогодні веб-сайти все більше схожі на багатокористувацькі програми, а не статичні сторінки, які були близько 10 років тому. З одної сторони, причина полягає в тому, що користувачі намагаються отримувати та створювати інформацію відповідно до своїх особистих характеристик та потреб.

З іншої сторони, власники веб-сайтів сподіваються надати користувачам більш зручний інтерфейс, обробки інформації. Веб-сайт має інтерфейс (front-end) та бізнес-логіку (back-end) [7].

Сьогодні існує два основних способи створення веб-сайтів: більш сучасним є багатосторінкова програма (MPA) - більш традиційний спосіб і односторінковий додаток (SPA).

Багатосторінкові програми (MPA) складаються з декількох сторінок зі статичною інформацією (зображення, текст) та посилань на інші сторінки з однаковим вмістом.

При переході на іншу сторінку, браузер робить новий запит до сервера і перезавантажує всі ресурси, навіть ті компоненти, які повторюються на всіх сторінках (наприклад, колонтитули, заголовки). Тому продуктивність витрачається на завантаження одних і тих же предметів, це впливає на швидкість і продуктивність.

Основними технологіями таких сайтів є HTML і CSS. Ці методи використовувались для розробки найдавніших веб-сайтів і продовжують використовуватися сьогодні для створення сучасних веб-сайтів [9]. Принцип дії цього методу показаний на рисунку 1.1.

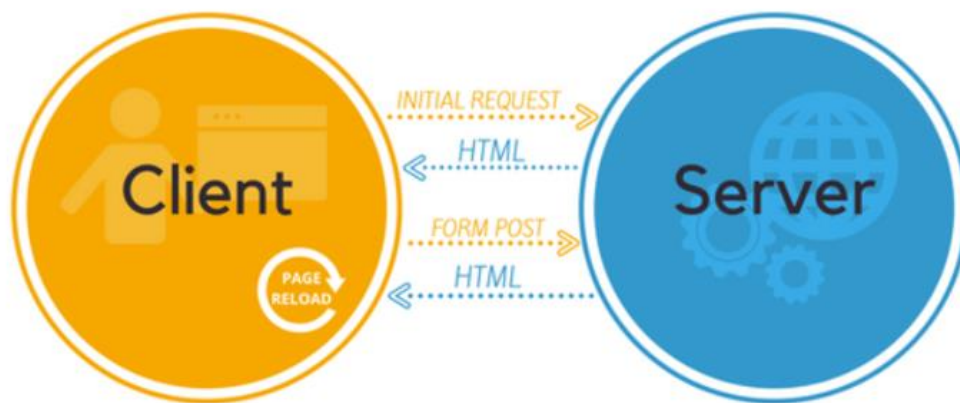


Рисунок 1.1 – Принцип роботи багатосторінкового додатку

Більш складні інтерактивні компоненти (наприклад, форми) можуть знадобитися при розробці веб-сайту. У цьому випадку все ще можна використовувати традиційні методи проектування, але стандартні можливості технології HTML будуть обмежені. Однак для нестандартних, складних та інтерактивних речей існують більш підходящі технології, такі як JavaScript та AJAX. Завдяки розвитку цих технологій став можливим розвиток Web 2.0. Сьогодні ці методи можна використовувати для створення складних фігур, різноманітних анімацій, діаграм та динамічно оновлюваної графіки без необхідності перезавантажувати всю сторінку, а лише змінені частини сторінки [10].

Недоліки багатосторінкових додатків: потрібно більше часу на розробку веб-сайтів для настільних ПК та мобільних пристроїв, а додавання нових функцій до програми збільшить вартість змін.

Переваги багатосторінкових додатків [14]:

1. Стек технологій, необхідний для розробки багатосторінкових програм, менший, тому вартість таких додатків дешевша.
2. Багатосторінкова архітектура (МРА) дозволяє легко оптимізувати кожен сторінку для пошукових систем. Розробники можуть додавати мета-теги на будь-яку сторінку.

Односторінковий додаток (SPA) – це програма, яка працює в браузері та надає користувачам досвід, близький до використання настільного додатка. Його архітектура побудована таким чином, що при переході на нову сторінку буде оновлено лише частина вмісту. Тому немає необхідності перезавантажувати ті самі елементи. Це дуже зручно для користувачів та розробників. Прикладом технології SPA є сервіс Gmail, впроваджений компанією Google. Для розробки SPA використовується одна з найпопулярніших мов програмування - JavaScript. На рисунку 1.2 показано принцип роботи односторінкової програми.

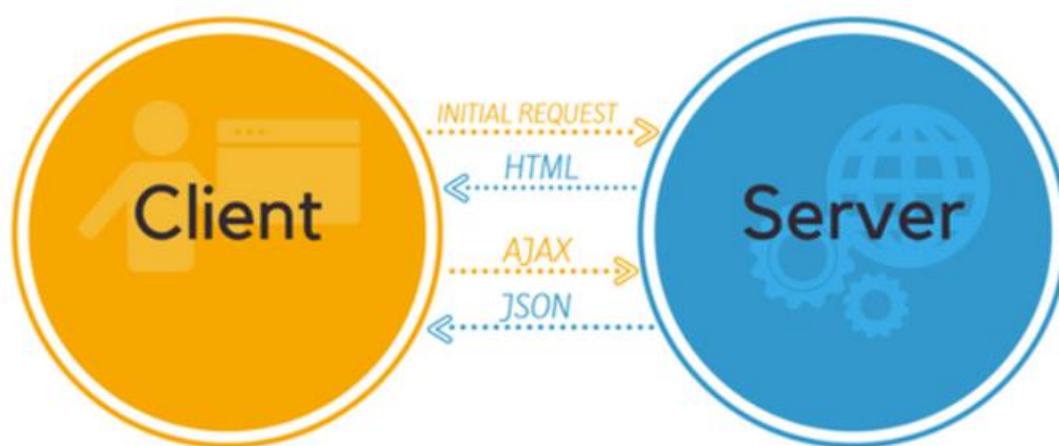


Рисунок 1.2 – Принцип роботи односторінкової програми [9]

За допомогою бібліотеки jQuery може бути зроблено невеликий веб-додаток. Але слід зазначити, що jQuery дуже погано впливає на розробку великих проєктів.

По-перше, це пов'язано зі структурою коду. Для створення складних проєктів рекомендується використовувати більш потужний фреймворк, такий як React, Angular або Vue. Їх архітектура дозволяє створювати гнучкі веб-додатки, крім того, вони мають велику кількість готових рішень, що значно спрощують процес розробки. Крім того, можна створювати повноцінні мобільні додатки на основі цих фреймворків [18].

Недоліки односторінкових додатків [2]:

1. Продуктивність та оптимізація. Потрібно пам'ятати про мобільні пристрої та низькопродуктивні пристрої, оскільки односторінкова програма використовує великий JavaScript.

Також необхідно пам'ятати про розмір веб-додатку. Якщо проєкт великий, його потрібно розділити на окремі компоненти з динамічним завантаженням, щоб збільшити швидкість завантаження додатку.

2. Погана SEO оптимізація. SPA працює на основі JavaScript, завантажуючи інформацію відповідно до запиту клієнта. Пошукові системи навряд чи будуть імітувати таку поведінку. Оскільки більшість сторінок взагалі не можуть бути проскановані пошуковими системами.

3. Час завантаження тривалий, оскільки важкі клієнтські фреймворки повинні спочатку завантажитися у браузер.

Переваги односторінкових додатків [2]:

1. Висока швидкість розробки. Готові фреймворки та бібліотеки надають потужні інструменти та велику кількість готових рішень для розробки веб-програм.

2. Висока швидкість. Оскільки SPA не оновлює всю сторінку, а лише необхідні частини, це значно покращує швидкість роботи додатку.

Мобільні додатки. SPA дозволяють створювати мобільні додатки, оскільки розробники можуть повторно використовувати той самий внутрішній код для веб-програм та власних мобільних додатків.

SPA може ефективно кешувати будь-які дані. Програма лише надсилає запит, кешує всі дані, а потім може використовувати їх і навіть може працювати в автономному режимі.

Односторінкові веб-сайти чудово підходять для створення динамічних платформ з невеликими обсягами даних. Крім того, веб-програми на одній сторінці придатні для розробки мобільних додатків [11].

Порівняльну характеристику відомих рішень багатосторінкових та односторінкових веб-сайтів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика відомих рішень багатосторінкових та односторінкових веб-сайтів

	Багатосторінковий веб-сайт (МРА)	Одно-сторінковий веб-сайт (SPA)
Час на завантаження	+	–
Швидкість розробки	–	+
Продуктивність та оптимізація	+	–
Стек технологій	+	–
SEO Оптимізація веб-сайту	+	–

Кожна архітектура має свої переваги та недоліки і дуже підходить для конкретних типів проєктів. SPA відомий своєю швидкістю та можливістю розробляти мобільні додатки на основі готового коду. Але в той же час SEO-оптимізація пошукової системи SPA не є хорошою. Тому ця архітектура є хорошим методом для соціальних мереж та закритих спільнот, де оптимізація пошукових систем не важлива. МРА більше підходить для інтернет-магазинів, комерційних веб-сайтів, каталогів тощо.

Добре оптимізований МРА має високу швидкість, але все одно не дозволяє легко розробляти мобільні додатки. МРА та SPA з правильною архітектурою дуже підходять для розробки масштабованих веб-додатків [4].

Серед знайдених веб-сайтів на ринку можна виділити такі:

- сервісний центр «Пріорітет»;
- сервісний центр «Екватор»;
- сервісний центр «Контакт сервіс».

Сервісний центр – це організація, яка займається ремонтом техніки та послугами підтримки. Деякі сервісні центри включають гарантію та після гарантійний ремонт.

Сервісний центр ремонту електроніки – організація, або приватне підприємство, яке займається у пошуках несправності гаджетів і електроніки. Головна задача такої організації – усунути дисфункцію апарату і не допустити нових, забезпечити все необхідне, щоб апарат зміг виконувати свою функцію у власника як найдовше.

На рисунку 1.3 зображено сервісний центр «Пріоритет». На даному веб-сайті ми зможемо:

- переглянути розташування даного сервісу;
- послуги які надає сервіс (ремонт та модернізацію ноутбуків, ремонт та модернізацію системних блоків, ремонт оргтехніки, заправка та регенерація катриджів, ремонт та налаштування планшетів, смартфонів);
- авторизуватись;
- контактні дані;
- документ, де вказано ремонт техніки;
- статус замовлення.

Сервісний центр «Пріоритет», є багатосторінковим веб-сайтом, в якому присутня реєстрація користувачів.

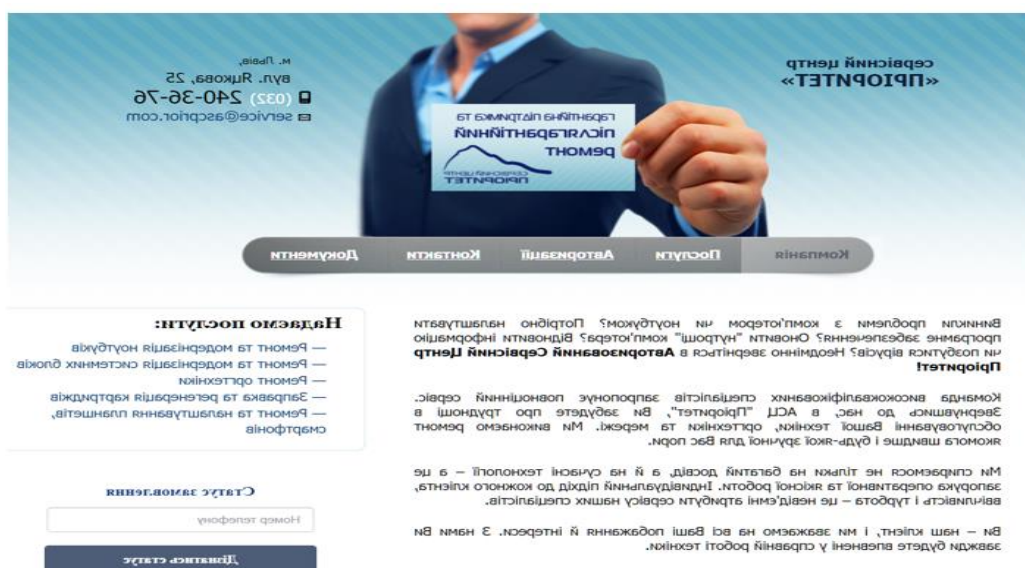


Рисунок 1.3 – Сервісний центр «Пріоритет»

На рисунку 1.4 зображено сервісний центр «Екватор», який займається ремонтом та обслуговуванням комп'ютерів, ноутбуків та іншої офісної техніки. На даному веб-сайті зображено:

- контакти, за якими можна зв'язатись із сервісним центром;
 - послуги які надає сервісний центр (ремонт ноутбуків, ПК, клавіатури)
- також надається гарантія на 24 місяці;
- виїзд майстра;
 - консультація з майстром.

Сервісний центр «Екватор» є багатосторінковим веб-сайтом.



Рисунок 1.4 – Сервісний центр «Екватор»

На рисунку 1.5 зображено сервісний центр «Контакт сервіс». Веб-сайт, на якому вказані послуги ремонту мобільних телефонів, а також можна переглянути таку інформацію як:

- діяльність;
- про нас;
- сервіс (заміна скла, оновлення програмного забезпечення, заміна акумулятора, заміна дисплею модуля).

Сервісний центр «Контакт сервіс» є багатосторінковим веб-сайтом.

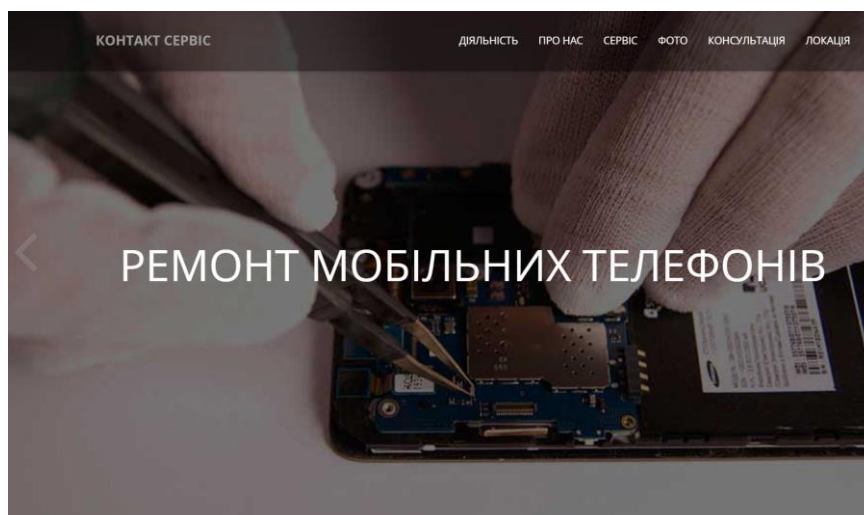


Рисунок 1.5 – Сервісний центр «Контакт сервіс»

Порівняльну характеристику відомих рішень подано в таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика відомих рішень

	Сервісний центр «Пріорітет»	Сервісний центр «Екватор»	Сервісний центр «Контакт сервіс»
Багатосторінковий веб-сайт (МРА)	+	+	+
Наявність реєстрації	+	—	—
Подання інформації про сервіс	+	—	+
Форма зворотнього зв'язку	+	—	—
Мобільний варіант веб-сайту	—	+	—

Продовження таблиці 1.2

	Сервісний центр «Пріоритет»	Сервісний центр «Екватор»	Сервісний центр «Контакт сервіс»
Зручність інтерфейсу	+	+	—
Визначення ОС пристрою	—	—	—
Цільова реклама	—	—	—
Інформаційний блок новин про техніку	—	—	—

Оскільки на сьогодні відомі рішення володіють такими недоліками як: багатосторінковий веб-сайт, наявність реєстрації, подання інформації про сервіс, форма зворотнього зв'язку, мобільний варіант веб-сайту, визначення ОС пристрою, цільова реклама, інформаційний блок новин про техніку, тому розробка веб-сайту для сервісу з комп'ютерної техніки «ПК Майстер» буде актуальною для розроблення.

Метою кваліфікаційної роботи є розробка веб-сайту для сервісу з комп'ютерної техніки «ПК Майстер». Для досягнення мети розробки необхідно розв'язати наступні задачі:

- здійснити критичний аналіз відомих рішень;
- провести огляд програмного забезпечення;
- розробити структуру веб-сайту та структуру бази даних;
- здійснити програмування веб-сайту;
- провести тестування веб-сайту та інструкцію щодо його обслуговування;

1.2 Огляд програмного забезпечення та мов програмування

1.2.1 Програмне забезпечення MySQL та сервер Apache

MySQL – це безкоштовна реляційна система управління базами даних, розроблена компанією «ТсХ» для збільшення швидкості обробки великих баз даних. Ця система управління базами даних із відкритим кодом (СУБД) була створена як альтернатива комерційним системам. MySQL з самого початку дуже схожий на mSQL, але з часом він продовжував розширюватися, і зараз MySQL є однією з найпоширеніших систем управління базами даних. В основному він використовується для створення динамічних веб-сторінок, оскільки має хорошу підтримку різних мов програмування.

Можливість MySQL – це компактний багатопотоковий сервер баз даних. Характеризується швидкою швидкістю, хорошою стабільністю та простотою використання. MySQL вважається хорошим рішенням для малих та середніх додатків.

Вихідний код сервера компілюється на багатьох платформах. Найповніша функція сервера знаходиться в системі UNIX, яка підтримує багатопотоковість, покращуючи тим самим продуктивність усієї системи. Функції сервера MySQL: простота установки та використання; підтримка необмеженої кількості користувачів для одночасного використання бази даних; кількість рядків у таблиці може досягати 50 мільйонів; швидка швидкість виконання команди; наявність простої та ефективної системи безпеки.

Apache – це веб-сервер HTTP. Apache Web Server – це міжплатформлене програмне забезпечення. Завдяки великій кількості документації та хорошій інтеграції зі стороннім програмним забезпеченням, Apache став дуже поширеним явищем. Підтримуються такі операційні системи: Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Переваги веб-сервера Apache:

- підтримка мов програмування, таких як PHP, Python, Ruby, Perl, ASP;

- просто підключається до зовнішніх модулів; підтримує технології PHP, CGI та FastCGI;
- забезпечує безпеку доступу до даних та наявність механізмів розмежування; можливість використання бази даних для автентифікації користувачів;
- гнучка та надійна конфігурація системи;
- підходить для додатків, які потребують надійного зашифрованого захисту даних;
- можливість створити власний каталог для веб-сайту;
- можливість налаштування віртуальних хостів, ви можете створити кілька віртуальних хостів на одному фізичному сервері;
- веде звіт протоколів того, що відбувається на сервері.

1.2.2 Мова програмування PHP

PHP (PHP: Hypertext Preprocessor-PHP: Hypertext Preprocessor), раніше відомий як: Personal Personal Page Tools – мова програмування сценаріїв, створена для створення HTML-сторінок на стороні веб-сервера. PHP – одна з найбільш часто використовуваних мов програмування у веб-розробці (також Java, .NET, Perl, Python, Ruby). Більшість постачальників послуг хостингу підтримують PHP. PHP – це проєкт програмного забезпечення з відкритим кодом [29].

PHP інтерпретується веб-сервером як HTML-код, який передається клієнту. На відміну від мови сценаріїв JavaScript, користувачі не бачать PHP-код, оскільки браузер отримує заповнений html-код. Це перевага безпеки, але це послабить інтерактивність сторінки. Але ніхто не забороняє використовувати PHP для генерації коду JavaScript. PHP – це скриптова мова програмування, де можна безпосередньо вбудувати HTML сторінки в код, і тоді інтерпретатор PHP правильно оброблятиме код. Обробник PHP просто починає виконувати код

після відкриваючого тегу (<? Php) і продовжує виконувати, поки не зустріне закриваючий тег (?>) [25].

Різні функції PHP дозволяють уникнути написання багаторядкових функцій для користувачів, як і те, що відбувається в C або Pascal.

Наявність багатьох інтерфейсів баз даних.

PHP має вбудовані бібліотеки для обробки MySQL, PostgreSQL, mSQL, Oracle, DBM, Hyperware, Informix, InterBase та Sybase. За допомогою Open Database Connectivity Standard (ODBC) можна підключатися до всіх баз даних, що мають драйвери.

PHP здається знайомим програмістам, що працюють у різних сферах. Багато мовних структур запозичені у C та Perl. Код PHP дуже схожий на код у типовій програмі на мові C або Pascal. Це значно зменшує початкове навантаження під час вивчення PHP. PHP – це мова, яка поєднує в собі переваги Perl і C. Хоча PHP є досить молодого мовою, вона настільки популярна серед веб-програмістів, що зараз є найпопулярнішою мовою для створення веб-додатків (скриптів) [31].

Оригінальний код надається безкоштовно.

Стратегія з відкритим кодом та розповсюдження оригінальної програми серед громадськості, безсумнівно, позитивно вплинули на багато проектів, насамперед Linux, хоча успіх проекту Apache сильно зміцнив джерело прихильників Open. Це стосується і історії PHP, оскільки підтримка користувачів з усього світу завжди була дуже важливим фактором у розробці PHP-проектів.

Крім того, користувачі PHP у всьому світі є колективною службою підтримки, і можна знайти відповіді навіть на найскладніші питання в популярних електронних конференціях [28].

При програмуванні для багатокористувацьких середовищ (включаючи Інтернет) ефективність є дуже важливим фактором. Важливою перевагою PHP є те, що ця мова інтерпретується. Це дозволяє обробляти сценарії на досить високій швидкості.

За деякими оцінками, більшість PHP-скриптів (особливо скриптів, які не дуже великі) обробляються швидше, ніж аналогічні програми, написані на Perl. Однак незалежно від того, що робить розробник PHP, скомпільований виконуваний файл буде працювати швидше – десятки разів, іноді сотні разів. Але продуктивності PHP досить для створення цілком серйозних веб-додатків.

1.2.3 Мова програмування JavaScript

JavaScript (JS) – це динамічна, об'єктно-орієнтовна мова програмування. Впровадження стандарту ECMAScript. Найчастіше використовується для створення веб-сценаріїв, які дозволяють клієнтові (пристрою кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними із сервером та змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікується як прототип (підмножина об'єктно-орієнтованих) з динамічно набраною мовою сценаріїв. На додаток до прототипів, JavaScript також частково підтримує інші парадигми програмування (імперативні та частково функціональні) та деякі супутні архітектурні особливості, включаючи: динамічне та слабе введення тексту, автоматичне управління пам'яттю, успадкування прототипу та функції як першокласні об'єкти.

JavaScript використовується для:

- Сценаріїв веб-сторінки, щоб вона стала інтерактивною;
- створення односторінкових та прогресивних веб-додатків (React, AngularJS, Vue.js);
- серверного програмування (Node.js (Express.js));
- виправлення програм (Електроніка, NW.js);
- мобільних додатків (React Native, Cordova);
- сценаріїв у додатку (наприклад, у програмі від Adobe Creative Suite або Apache JMeter);

- у середині документа PDF.

На даний момент JavaScript є однією з найпопулярніших мов програмування в Інтернеті. На перших порах її існування, більшість професійних програмістів скептично ставилися до мови, а цільовою аудиторією були програмісти-аматори [16].

Крім того, JavaScript містить набір вбудованих операцій, які контролюють логіку виконання програми. Граматика JavaScript в основному відповідає граматиці мови Java (тобто вона остаточно успадкована від C), але вона спрощена в порівнянні з нею, що полегшує вивчення мови сценаріїв.

Оскільки JavaScript є інтерпретованою мовою програмування, не існує строгого типу, він може працювати в різних середовищах, кожне середовище має свої функції сумісності, програміст повинен бути обережним і перевіряти свій код у різних можливих конфігураціях, чи працює він як слід.

1.2.4 Мова розмітки гіпертексту HTML

HTML (мова гіпертекстової розмітки) – це мова розмітки, яка використовується для розмітки веб-сторінок в Інтернеті. Браузер отримує документи HTML з веб-сервера або локального сховища та передає їх на мультимедійні веб-сторінки. HTML семантично описує структуру веб-сторінки, спочатку включаючи сигнали про зовнішній вигляд документа.

Елементи HTML – це будівельні блоки HTML-сторінок. Використовуючи структуру HTML, зображення та інші об'єкти (наприклад, інтерактивні форми) можуть бути вбудовані у відтворену сторінку.

HTML можна вбудовувати в програми, написані мовами сценаріїв (наприклад, JavaScript), які впливають на поведінку та вміст веб-сторінок. Увімкнення CSS визначає зовнішній вигляд і макет вмісту.

Засоби реалізації HTML використовуються для:

- створення структурованого документа, вказуючи структуру тексту: заголовки, абзаци, списки, таблиці, цитати;

- отримання інформації з Інтернету за допомогою гіперпосилань;
- створення інтерактивних форм;
- додавання до тексту зображення, звуків.

1.2.5 Класичні таблиці стилів CSS

CSS (Cascading Style Sheet) – це спеціальна мова стилів сторінок, що використовується для опису її зовнішнього вигляду. Сама сторінка написана мовою розмітки. CSS – це основна технологія всесвітньої павутини, поряд з HTML та JavaScript.

CSS найчастіше використовується для візуального відображення сторінок, написаних у форматі HTML та XHTML, але формат CSS може застосовуватися і до інших типів XML-документів.

CSS має різні рівні та файли конфігурації. Наступний рівень CSS створюється на основі попереднього CSS, додаючи нові функції або розширюючи існуючі функції. Рівні представлені як CSS1, CSS2 та CSS3. Файл конфігурації – це набір одного або декількох рівнів правил CSS, створених для певного типу пристрою чи інтерфейсу. Наприклад, є файли конфігурації CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадне або блок-макет) замінив макет таблиці веб-сторінок. Головною перевагою макета блоку є відокремлення вмісту сторінки (даних) та його візуальна презентація.

Створювачі та відвідувачі веб-сторінок використовують CSS для визначення кольорів, шрифтів, макета та інших аспектів зовнішнього вигляду сторінки. Однією з головних переваг є можливість відокремлення вмісту сторінки (або вмісту, вмісту, як правило, HTML, XML або подібної мови розмітки) від типу документа (описаного в CSS).

1.3 Постановка задачі

Основне призначення проєкту – розробити веб-сайт для сервісу комп’ютерної техніки «ПК Майстер», на якому буде:

- розміщена інформація щодо послуг, які надаються сервісом;
- передбачатися можливість показу реклами в залежності від типу пристрою, на якому переглядається веб сайт;
- реклама продукції, послуг, ідей. Правильно зроблений веб-сайт із легкістю приведе клієнта до висновку про необхідність покупки товару, або послуг, що пропонуються на ньому;
- забезпечуватися зворотній зв’язок з клієнтами;
- безкоштовне надання інформації;
- підтримка клієнтів.

Для створення даного сайту буде використано такі мови програмування:

PHP – тому що одна з найбільш часто використовуваних мов програмування у веб-розробці.

JavaScript – тому що найчастіше використовується для створення веб-сценаріїв, які дозволяють клієнтові (пристрою кінцевого користувача) взаємодіяти з користувачем, керувати браузером, синхронно обмінюватися даними із середовищем та змінювати структуру і зовнішній вид сторінки.

CSS – тому що це спеціальна мова стилів сторінок, що використовується для опису її зовнішнього вигляду. CSS – використовується для візуального відображення сторінок написані у форматі HTML. Також формат CSS може застосовуватися і до інших типів XML-документів.

HTML – тому що це мова розмітки, яка використовується для розмітки веб-сторінок в інтернеті. HTML – симетрично описує структуру веб-сторінки, спочатку включаючи сигнали про зовнішній вигляд документа.

Для виконання поставлених вимог завдання потрібно встановити та сконфігурувати на сервері відповідне програмне забезпечення:

- ОС Ubuntu Linux 18.04;

- MySQL 5.7 або вище;
- Apache 2.4 або вище;
- PHP 7.2 x64;
- файрвол iptables.

Документація на програмний продукт повинна включати в себе [18]:

- інструкції з налаштування програмного середовища (хостингу);
- інструкції з розробки ПЗ та його структура.

Вся документація має бути в зрозумілому вигляді потрібному для коректної взаємодії користувача з програмним продуктом.

Розробка веб-сайту повинна бути проведена в такі етапи [19]:

- 1) розробка структури веб-сайту;
- 2) проектування;
- 3) розробка клієнтської частини;
- 4) розробка серверної частини;
- 5) розробка програмної документації;
- 6) тестування сайту.

Розробка веб-сайтів – це комплексний багатоетапний процес, що вимагає знання безлічі різних технологій і мов програмування, вміння працювати з базами даних, використовувати безліч інструментальних засобів і програмних пакетів [27].

1.4 Висновок до першого розділу

В першому розділі кваліфікаційної роботи проведено аналіз відомих рішень, порівняльну характеристику односторінкових та багатосторінкових веб-сайтів, їхні переваги та недоліки, також проведено порівняльну характеристику відомих рішень веб-сайтів сервісних центрів, розглянуто та вибрано програмне забезпечення: MySQL, Apache, та мови програмування: PHP, JavaScript, мови розмітки гіпертексту HTML, класичні таблиці стилів CSS, складено постановку задачі.

РОЗДІЛ 2. РОЗРОБКА ВЕБ-САЙТУ ДЛЯ СЕРВІСУ З РЕМОНТУ КОМП'ЮТЕРНОЇ ТЕХНІКИ «ПК МАЙСТЕР»

2.1 Розробка структури веб-сайту та структури бази даних

Метою даного дипломного проєкту є розробка веб-сайту для сервісу з ремонту комп'ютерної техніки «ПК Майстер». Оскільки даний сайт є багатосторінковим, то доцільно буде скласти структурну схему сайту, яку наведено на рисунку 2.1, а також файлову систему сайту.

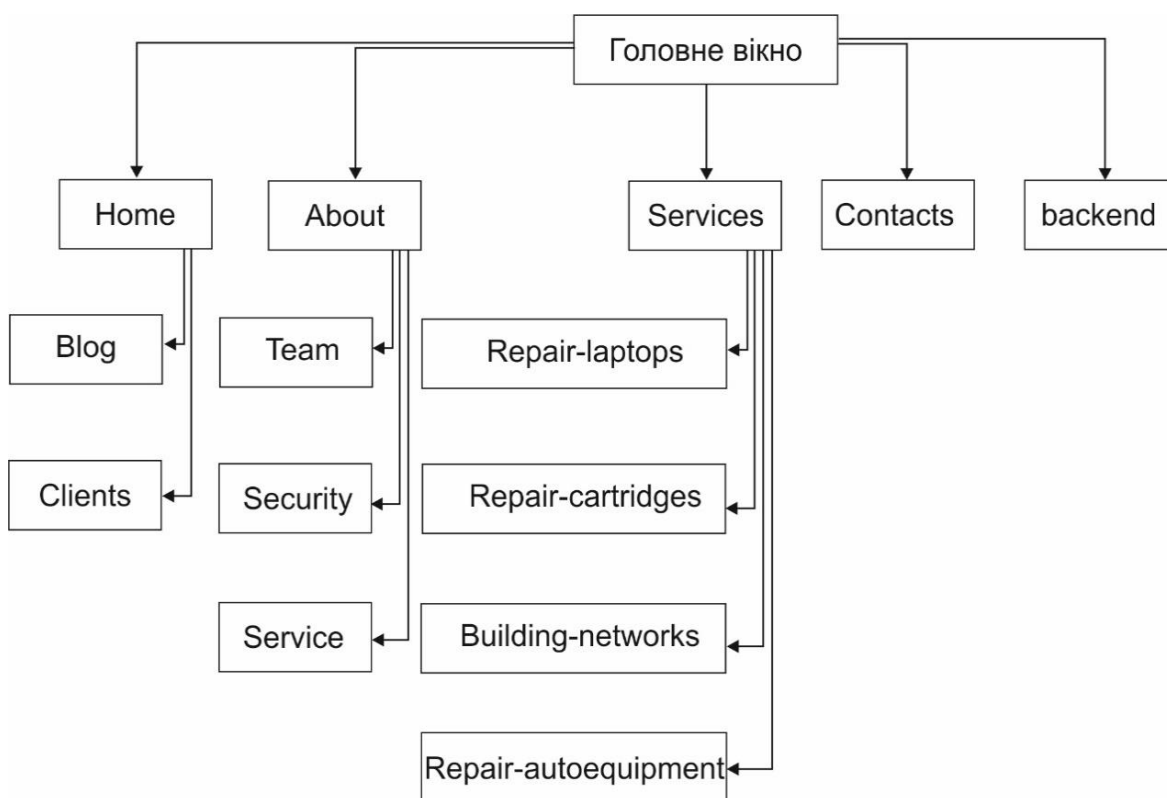


Рисунок 2.1 – Структурна схема сайту

Веб-сайт передбачає два рівні доступу:

- 1) адміністратор;
- 2) користувач.

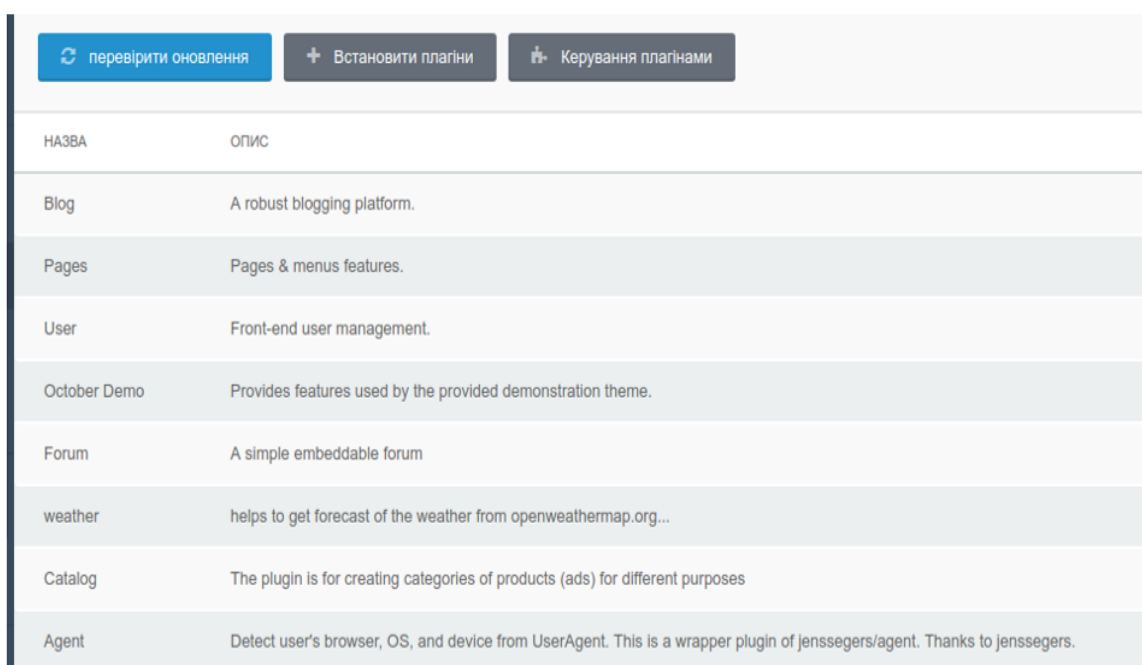
Звичайний користувач, тобто звичайний відвідувач зможе [25]:

- переглядати сторінки сайту призначені для нього;

- реєструватись;
- авторизуватись для входження в особистий кабінет;
- переглядати інформацію у своєму профілі та змінювати її.

У користувача з правами адміністратора буде на порядок більше функціоналу включно з правами звичайного користувача, а саме, у нього буде можливість керувати вмістом сайту, користувачами, їх правами.

Сайт побудовано з використанням October CMS. Для даної системи керування контентом розроблено шаблон (тему) та два плагіни. Створені плагіни можуть бути додані в систему шляхом копіювання їх у відповідну папку October CMS та викликом відповідної консольної команди `php artisan`. Один із плагінів призначений для визначення типу платформи користувача, що переглядає веб-сайт. Інший плагін призначений для створення каталогу реклами, яка буде відображатися в залежності від платформи користувача з якої він переглядає веб-сайт, `october:up`, або через адміністративну панель October CMS, як показано на рисунку 2.2.



<div> <div>↻ перевірити оновлення</div> <div>+ Встановити плагіни</div> <div>⚙ Керування плагінами</div> </div>	
НАЗВА	ОПИС
Blog	A robust blogging platform.
Pages	Pages & menus features.
User	Front-end user management.
October Demo	Provides features used by the provided demonstration theme.
Forum	A simple embeddable forum
weather	helps to get forecast of the weather from openweathermap.org...
Catalog	The plugin is for creating categories of products (ads) for different purposes
Agent	Detect user's browser, OS, and device from UserAgent. This is a wrapper plugin of jenssegers/agent. Thanks to jenssegers.

Рисунок 2.2 – Встановлені плагіни

На даному рисунку показано встановлені два плагіни з іменами:

- Catalog – призначений для створення каталогу реклами.
- Agent – призначений для визначення платформи з якої користувач переглядає сайт, ОС, яку використовує користувач.

Розглянемо структуру шаблону.

Тема (шаблон) – це набір папок, які знаходяться за замовчуванням в папці /themes і можуть містити такі об'єкти:

- сторінки (Pages) – сторінки сайту;
- фрагменти (Partials) – містить повторно використовувані шматки HTML-розмітки;
- макети (Layouts) – визначає каркас сторінки;
- вміст (Content files) – текст, HTML або Markdown блоки, які можна редагувати незалежно від сторінки або макета;
- asset files – файли на зразок зображень, CSS і JavaScript файлів.

Структура шаблону Valtheme зображена на рисунку 2.3.

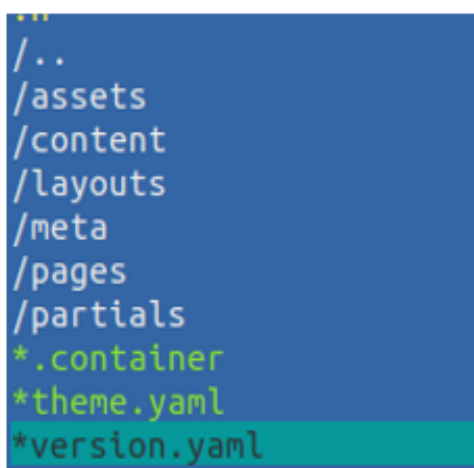


Рисунок 2.3 – Структура шаблону Valtheme

Дана тема може бути встановлена використовуючи адміністративну частину October CMS. На рисунку 2.4 показано встановлену тему Valtheme.

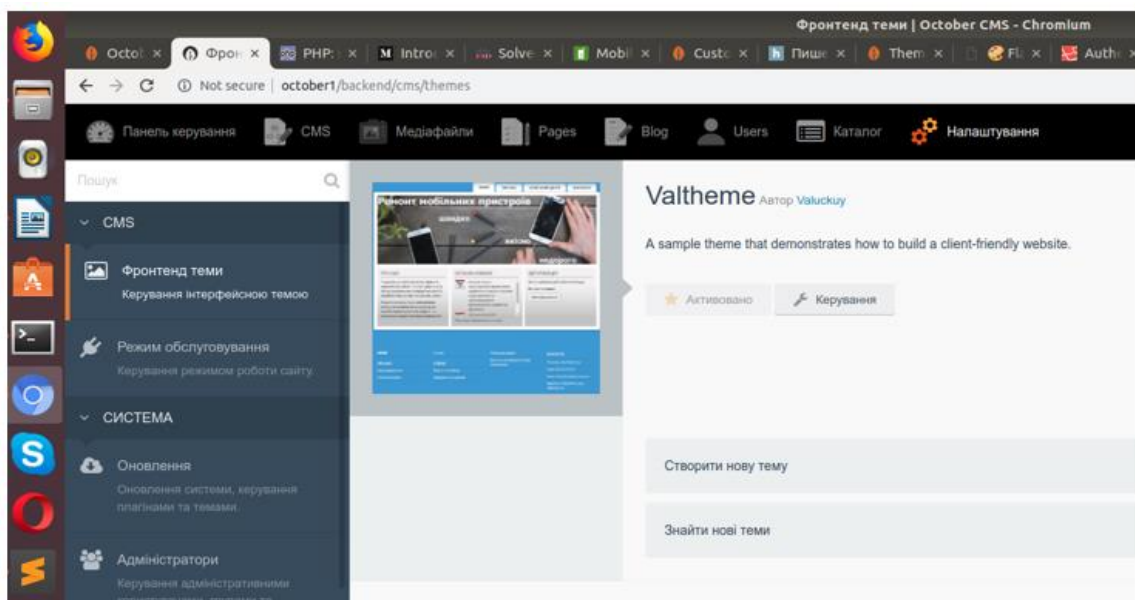


Рисунок 2.4 – Тема Valtheme

Кожна з сторінок наведених на рисунку 2.1 має у своїй основі шаблон. Шаблон визначає загальні елементи сторінки, які повторюються на кожній з сторінок.

Розроблено такі шаблони:

- client.htm (використовується для сторінок:);
- default.htm;
- home.htm;
- main-service.htm;
- service.htm.

Нижче, на лістингу 2.1 наведено перелік всіх сторінок та шаблонів, які вони використовують (вивід команди `grep -rin "layout = "`).

Лістинг 2.1 – Перелік всіх сторінок та шаблонів, які використовують «`grep -rin "layout = "`»

```
content/static-pages/building-networks.htm:4:layout = "service"
content/static-pages/clients-gallery.htm:4:layout = "client"
content/static-pages/services.htm:4:layout = "main-service"
content/static-pages/about.htm:4:layout = "default"
content/static-pages/repair-autoequipment.htm:4:layout = "service"
content/static-pages/about-team.htm:4:layout = "default"
content/static-pages/clients-newsletters.htm:4:layout = "client"
```

```

content/static-pages/about-services.htm:4:layout = "default"
content/static-pages/repair-cartridges.htm:4:layout = "service"
content/static-pages/repair-laptops.htm:4:layout = "service"
content/static-pages/about-facilities.htm:4:layout = "default"
content/static-pages/clients.htm:4:layout = "client"
content/static-pages/index.htm:4:layout = "home"
content/static-pages/clients-announcements.htm:4:layout = "client"
pages/404.htm:3:layout = "default"
pages/contact.htm:3:layout = "default"
pages/blog/category.htm:3:layout = "default"
pages/blog/post.htm:3:layout = "default"
pages/blog.htm:3:layout = "default"
pages/account.htm:3:layout = "default"
pages/forgot-password.htm:3:layout = "default"
pages/error.htm:3:layout = "default".

```

Розглянемо лістинг 2.2 структуру шаблону Nome з коментарями.

В цій секції вказуємо, які плагіни будемо використовувати. В нашому випадку ми використовуємо ряд плагінів October CMS та наш плагін Agent.

Лістинг 2.2 – Структура шаблону Nome

```

description = "Default layout"
[account]
[session]
[staticMenu mainMenu]
code = "main-menu"
[staticPage]
[Agent]

```

У наступному розділі 2.2 буде код РНР. Нижче наведений код визначає чи користувач використовує Windows платформу для перегляду веб-сайту і відповідно повертає масив банерів в шаблон, на основі якого буде генеруватися кінцева сторінка для перегляду користувачем.

Також дана секція коду визначає чи користувач зайшов на сайт з мобільного пристрою та повертає відповідний масив банерів. У випадку, якщо користувач не є Windows-користувачем або мобільним користувачем, то йому буде повернуто масив банерів з назвою “Common” (Див. лістинг 2.3).

Лістинг 2.3 – Визначає який пристрій використовує користувач

```

use RainLab\Pages\Classes\Page;

```

```

use Valuckuy\Catalog\Components\Products;
function onStart()
{
    $agent = new Agent();
    if ( $agent->platform() == 'Windows' ) {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(1);
    } elseif ( $agent->isPhone() ) {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(2);
    } else {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(3);
    }
}

```

(Див. Додаток Е – Лістинг коду) Наведена секція є власне шаблоном, на основі нього (Layout), фрагментів (Partials), вмістимого (Content) і буде сформовано кінцеву сторінку для користувача.

Наведемо приклад фрагменту Partial, який буде вбудовувати в сторінку відображення банеру з рекламою (Див. лістинг 2.4).

Лістинг 2.4 – Фрагмент Partial

```

{% if not bannerImage %}
    {% set bannerImage = banner ? banner|media :
    (random(randomImage)) %}
{% endif %}
<div class="title-image">

</div>

```

Його логіка наступна, у випадку передачі масиву (ім'я масиву randomImage) банерів один із них буде випадковим чином вибраний та переданий у шаблон. У випадку передачі одиничного банеру, він буде переданий одразу у шаблон.

Наступний фрагмент sideMenu генерує бокове меню з статичних сторінок (Див. лістинг 2.5).

Лістинг 2.5 – Фрагмент sideMenu

```

{% set parentOrSelf = page.parent ?: page %}
{% set childPages = parentOrSelf.children %}
<ul>

```



```

<li class="head-link {{ page.url == parentOrSelf.url ? 'active' }}">
<a href="{{ parentOrSelf.url|app }}">
{{ parentOrSelf.title }} </a>
</li>
    {% for childPage in childPages if not
childPage.navigation_hidden %}
<li class="{{ page.url == childPage.url ? 'active' }}">
<a href="{{ childPage.url|app }}">
{{ childPage.title }} </a>
</li>
    {% endfor %}
</ul>

```

Фрагмент Scripts для вставки у шаблон посилань на javascript фреймворки (Див. лістинг 2.6).

Лістинг 2.6 – Фрагмент Scripts

```

<script src="{{ [
    'assets/javascript/jquery.js',
    'assets/javascript/app.js'
]|theme }}"></script>
{% framework extras %}
{% scripts %}

```

Фрагмент meta для вставки тегів html <meta> (Див. лістинг 2.7).

Лістинг 2.7 – Фрагмент meta

```

<meta charset="utf-8">
<title>PC Repair Company - {{ this.page.title }}</title>
<meta name="description" content="{{ this.page.meta_description }}">
<meta name="title" content="{{ this.page.meta_title }}">
<meta name="author" content="RepairPC company">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="icon" type="image/png" href="{{ 'assets/images/october.png'|theme }}" />
<meta name="generator" content="RepairPC company" />
{% styles %}
{# Vendor #}
<link href="{{ [
    'assets/less/vendor.less',
]|theme }}" rel="stylesheet" />
{# Theme #}
<link href="{{ ['assets/less/theme.less']|theme }}" rel="stylesheet" />

```

Оскільки в основі створеного веб-сайту лежить October CMS, то більшість таблиць ER-діаграми будуть належати саме їй. У (Див. Додаток Б) “ER-діаграма сайту” наведено перелік таблиць, які використовуються вище згаданою системою керування контентом. Додатково до основних таблиць буде використано таблиці бази даних для плагіну Catalog. Структура описана в таблиці 2.1 та наведена на рисунку 2.5.

Таблиця 2.1 – Список полів таблиці valuckuy_catalog_products

Ім'я поля	Тип поля	Призначення
Id	Int	Ідентифікатор продукту (реклами)
title	Varchar	Латинська назва продукту (реклами)
description	Text	Опис
created_at	timestamp	Дана створення
updated_at	timestamp	Дата оновлення

Крім таблиці valuckuy_catalog_products буде використано систему таблицю system_files, яка буде зв'язана з вище описаною таблицею через зовнішні ключі, як показано на рисунку 2.5 відношенням «один до багатьох».



Рисунок 2.5 – ER-діаграма плагіну Catalog

Нижче наведено модель, яка буде застосована для побудови взаємозв'язків «один до багатьох» (Див. лістинг 2.8).

Лістинг 2.8 – Модель взаємозв'язків «один до багатьох»

```
<?php namespace Valuckuy\Catalog\Updates;
use Schema;
use October\Rain\Database\Updates\Migration;
class CreateProductsTable extends Migration
{
    public function up()
    {
        Schema::create('valuckuy_catalog_products', function($table)
        {
            Schema::dropIfExists('valuckuy_catalog_products');
            $table->engine = 'InnoDB';
            $table->increments('id');
            $table->string('title');
            $table->text('description');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('valuckuy_catalog_products');
    }
}
```

2.2 Програмування веб-сайту

Для веб-сайту сервісу з ремонту комп'ютерної техніки «ПК Майстер» буде розроблено плагін Catalog для створення реклами для різних користувачів, в залежності від того, яку вони ОС використовують та в залежності від того, з якого пристрою вони переглядають веб-сайт (мобільний пристрій чи стаціонарний ПК).

Для реєстрації плагіну створюємо наступний файл Plugin.php (Див. Додаток Ж – Плагін Catalog)

Клас Products.php, який буде виконувати функцію компонента наведено (Див. Додаток И – Agent).

Оскільки в основі структури системи керування контентом лежить архітектура MVC, то для плагіну також потрібно написати моделі, які наведено (Див. Додаток И – Плагін Agent).

Для визначення типу платформи, операційної системи користувача з яких він переглядає сайт буде використано плагін Agent (Див. лістинг 2.9).

Лістинг 2.9 – Плагін Agent

```
use RainLab\Pages\Classes\Page;
use Valuckuy\Catalog\Components\Products;
use Valuckuy\Agent\Classes\Agent;
function onStart()
{
    $agent = new Agent();
    if ( $agent->platform() == 'Windows' ) {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(1);
    } elseif ( $agent->isPhone() ) {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(2);
    } else {
        $o = new Products();
        $this['randomImage'] = $o->getImagesArray(3);
    }
}
?>
```

Визначення платформи та типу операційної системи за такими етапами:

- створюється об’єкт Agent;
- виклик методу platform об’єкту Agent;
- вивчається платформа (Windows, або мобільний пристрій);
- створення об’єкту Products;
- отримання масивів банерів за ідентифікацією.

На рисунку 2.6 наведено блок-схему алгоритму визначення платформи та типу операційної системи.

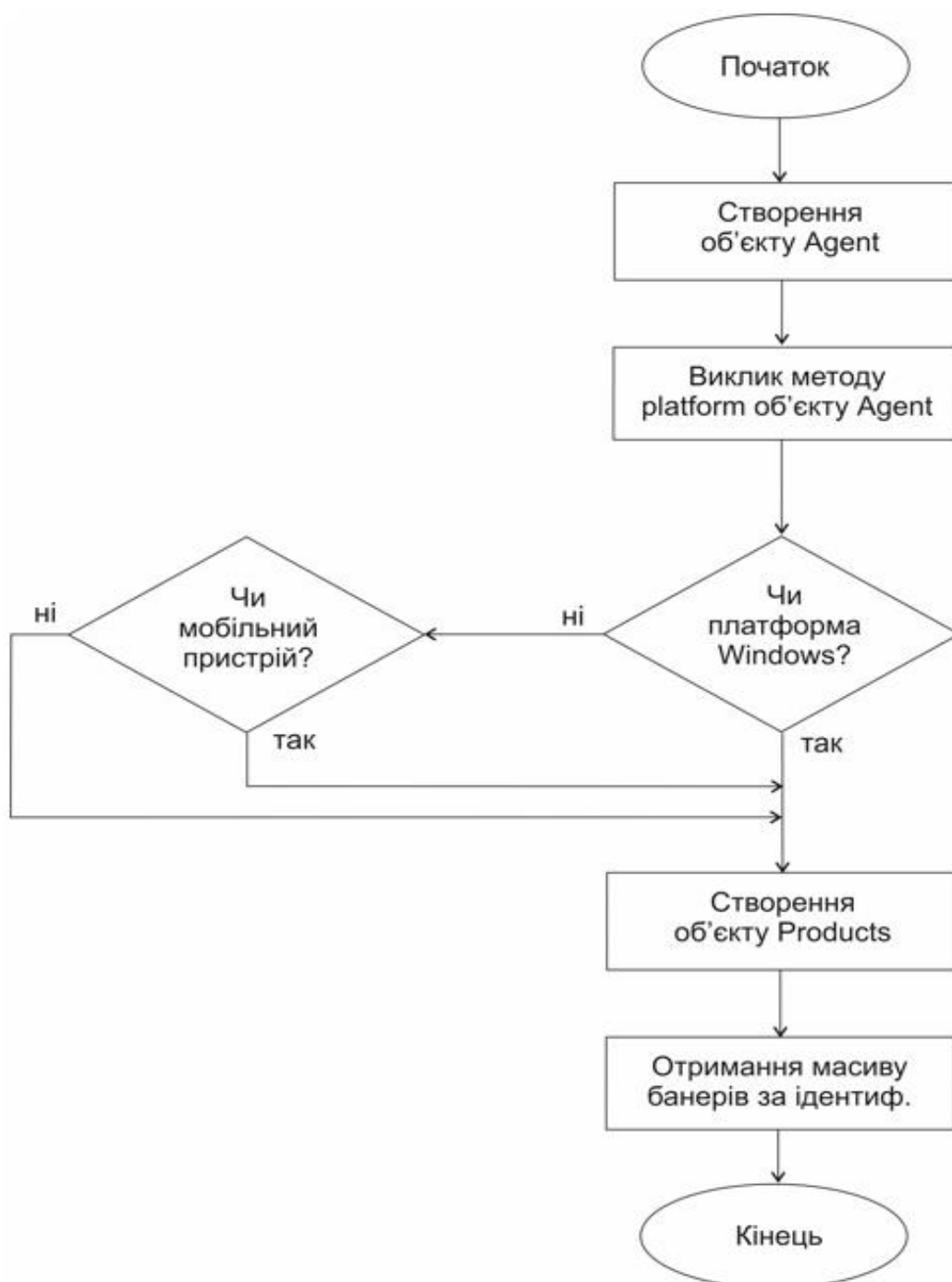


Рисунок 2.6 – Блок схема алгоритму визначення типу платформи та типу пристрою

Подана блок схема складається з функціональних блоків, які є структурно незалежними частинами автоматичної системи, яка виконує певні функції. Функціональні блоки на рисунку представлені прямокутниками, всередині яких надписують їх найменування відповідно до виконуваних функцій.

2.3 Тестування веб-сайту та інструкція щодо його обслуговування

Важливим етапом розробки веб-сайту є перевірка його працездатності сайту у вигляді спеціального тестування. Воно включає в себе такі етапи:

- перегляд сайту на різних платформах;
- перегляд сайту на кросбраузерність;
- перевірка часу завантаження сайту;
- перевірка властивостей кожної сторінки сайту: заголовків, ключових слів, описи або інших мета-тегів;

Для тестування сайту будуть використані такі браузери: Chrome, Opera, InternetExplorer, MozillaFirefox. На рисунку 2.7 показано приклад відображення сайту у Google Chrome з включеною можливістю відображення в мобільному пристрої (розширення 430x826) .



Рисунок 2.7 – Відображення сайту в браузері Google Chrome з мобільного пристрою

На рисунку 2.8 показано відображення сайту при розширенні 1920x1080.

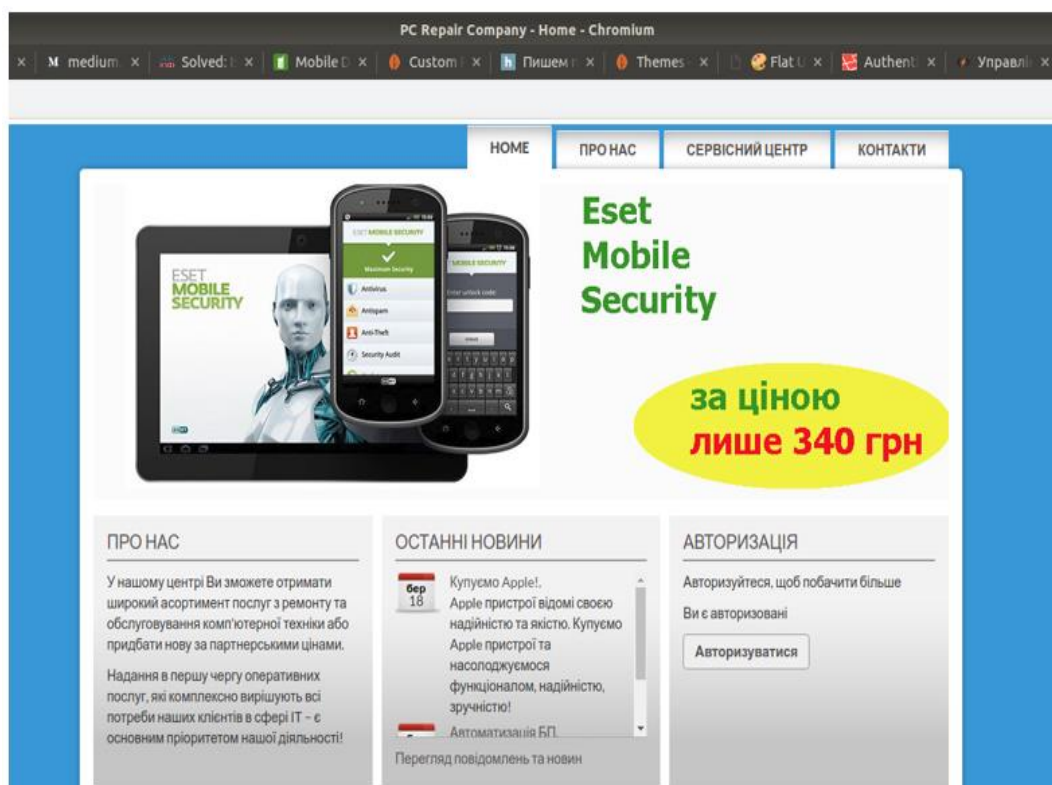


Рисунок 2.8 – Відображення сайту на Desktop-версії браузера

Крім вище описаних тестувань потрібно протестувати коректність роботи плагінів Catalog та Agent. Для цього потрібно встановити спеціальне розширення User Agent Switcher.

User Agent Switcher – це невелике розширення для Firefox, яке додає в панель браузера спеціальну кнопку, при натисканні якої зможемо змінити User Agent, яким видається браузером при відвідуванні сайтів.

За допомогою нього можна емулювати значення User Agent, які будуть передаватися на сервер, плагіни Catalog та Agent повинні забезпечити коректне переключення банерів в залежності від типу ОС користувача та платформи, з якої він переглядає сайт.

На рисунку 2.9 показано вигляд сайту при включеному режимі “Chrome on Windows”.

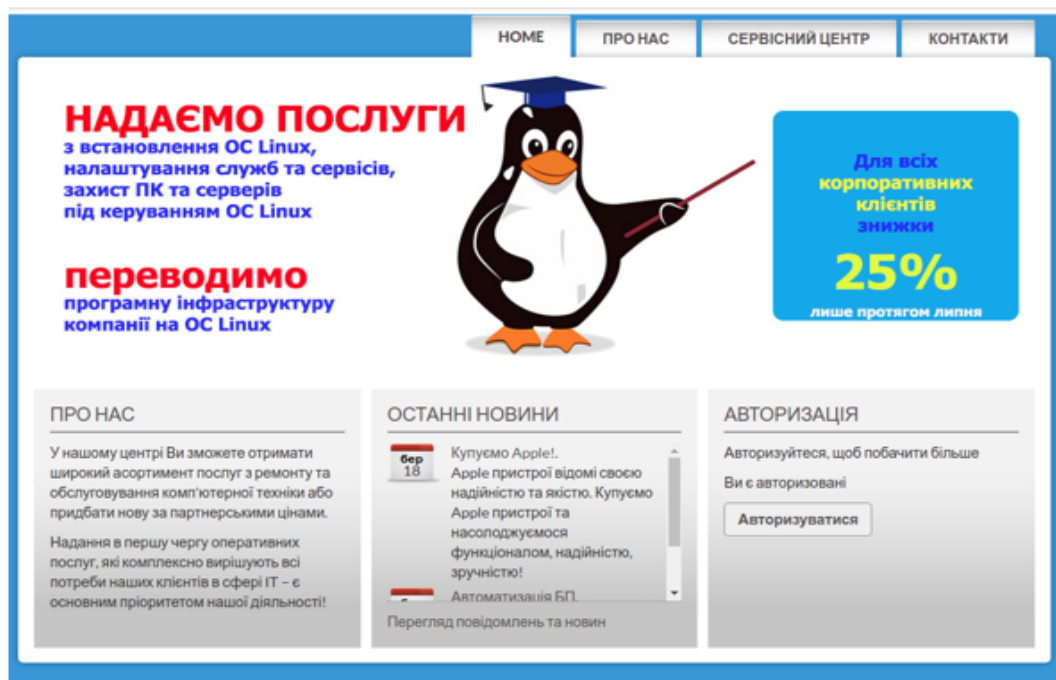


Рисунок 2.9 – Відображення сайту при включеному режимі «Chrome on Windows»

На рисунку 2.10 показано вигляд сайту при включеному режимі “Chrome on Windows”.

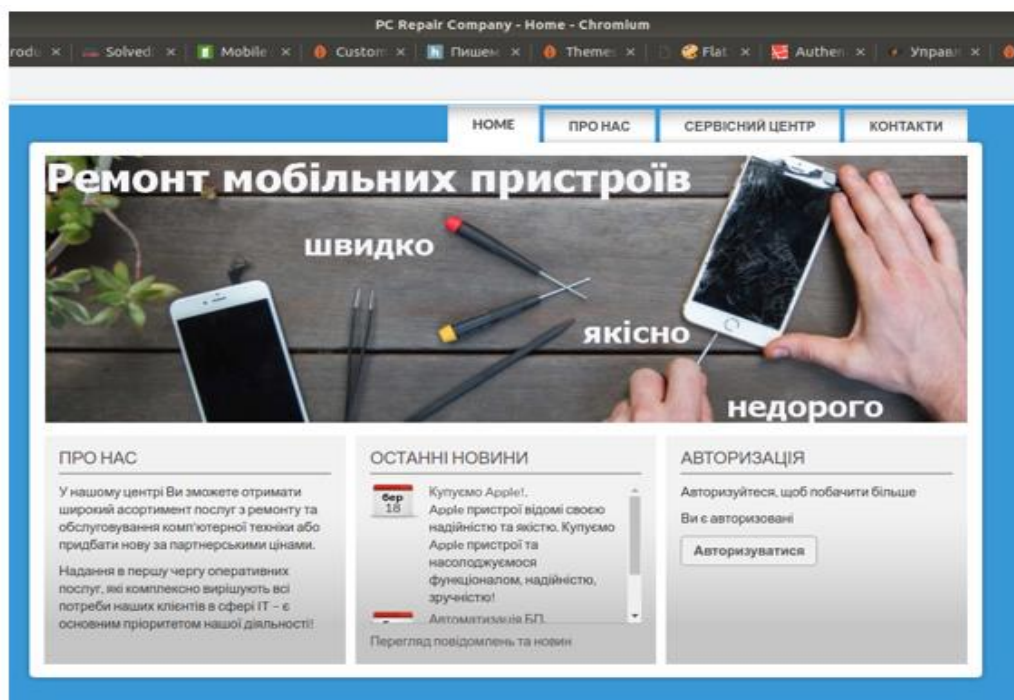


Рисунок 2.10 – Відображення сайту при включеному режимі «iPhone»

Веб-сайту для сервісу з ремонту комп'ютерної техніки «ПК Майстер» є звичайним сайтом для рекламування послуг фірми. Для відвідувачів сайту функціонал сайту буде звичайним як для інтернет-магазину. Користувач може переглядати інформацію про послуги, які надає фірма та замовляти їх. Для навігації по категоріях сайту можна вибрати верхнє меню або меню у footer`і сайту. Сторінка «Про нас» додатково має власне бокове меню, показане на рисунку 2.11.

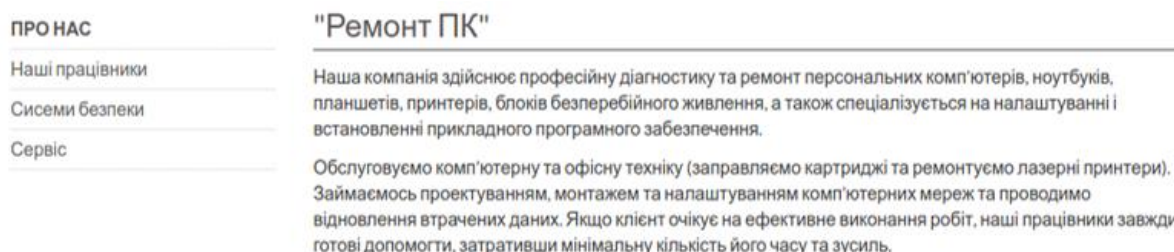


Рисунок 2.11 – Бокове меню «Про нас»

Користувачі мають власний Кабінет користувача, де вони можуть отримувати повідомлення, змінювати пароль. На рисунку 2.12 показано особистий кабінет користувача сайту.

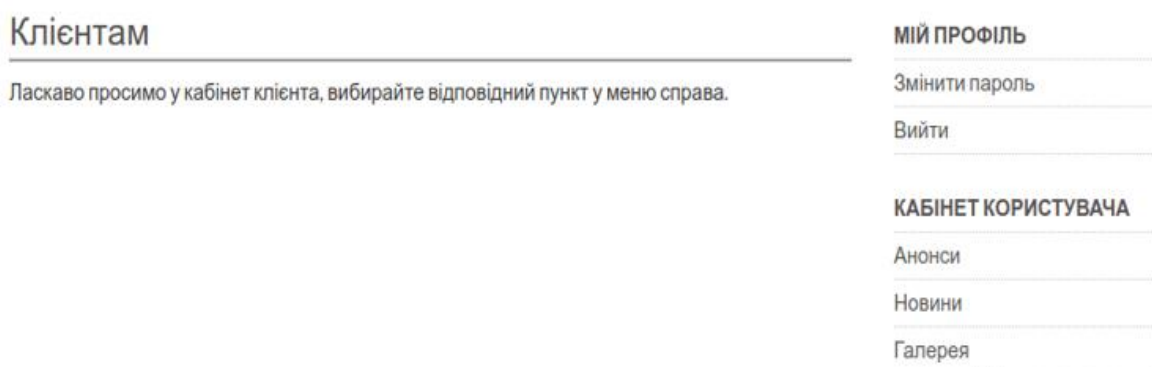


Рисунок 2.12 – Процес оформлення замовлення

Крім того для взаємодії з користувачем було розроблено форму відправки повідомлень, яку показано на рисунку 2.13.

Зателефонуйте нам

Потрібно інформацію про нашу фірму?

Просто заповни нашу форму вказавши всі необхідні дані та ваше запитання. Ми будемо раді відповісти на ваше запитання.

Телефон: 0675432112

Факс: 0352435678

Email: hello@repairpc.com.ua

Адреса м. Тернопіль, вул. Замкова 16

Онлайн запит

Ім'я *

Прізвище

Телефон *

Компанія

Email *

Повідомлення *

Відправити повідомлення!

Рисунок 2.13 – Форма відправки повідомлень

Для додавання нових банерів у відповідну колекцію банерів (Продукт) потрібно вибрати у верхньому меню адміністративної панелі розділ “Каталог” у якому відповідно вибрати кнопку «Новий список банерів», як показано на рисунку 2.14.

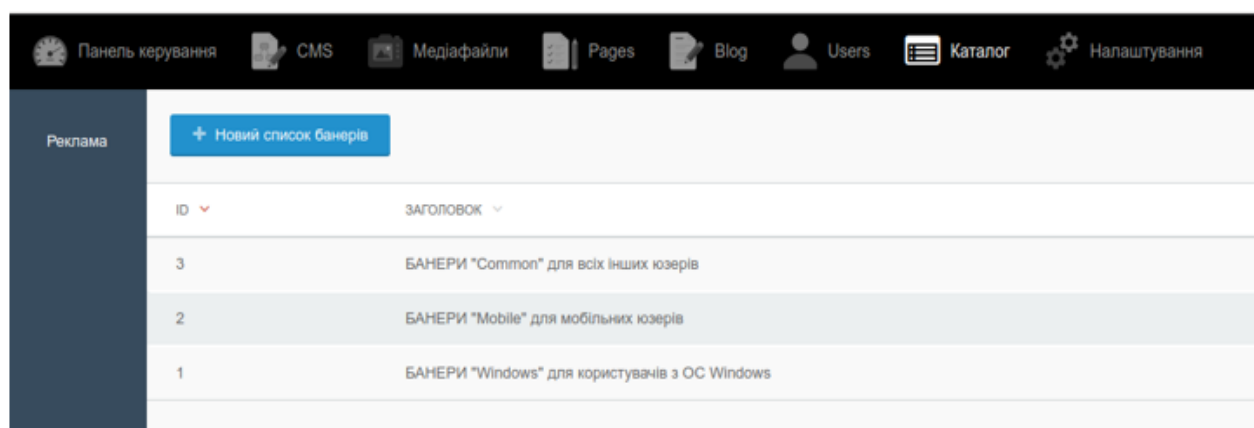


Рисунок 2.14 – Додавання нової колекції банерів

У вікні зображеному на рисунку 2.15 вказуємо назву колекції, опис детальний та додаємо власне банери, які будуть входити у колекцію.

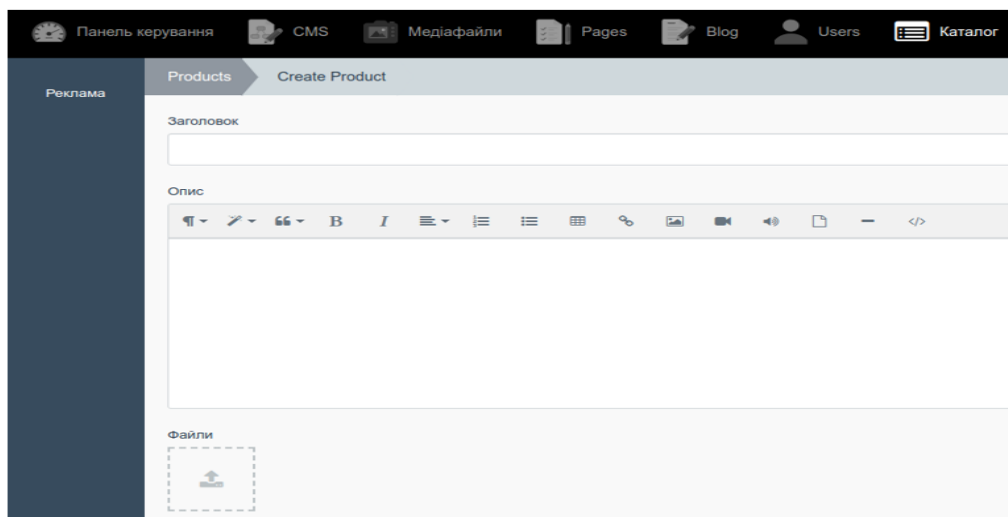


Рисунок 2.15 – Додавання банерів у колекцію

Інструкція з популяризації та підтримки сайту. Часи, коли для просування в топ достатньо було просто закупити побільше посилань, безповоротно закінчилися. Тренд 2021 року – комплексне просування сайту, тобто комплексна робота над внутрішніми і зовнішніми факторами, які впливають на ранжування в пошуку.

Тільки мультиканальність може забезпечити достатньою кількістю заявок. Для цього потрібно мати і просувати не лише сайт, але й групи в соцмережах, канали в месенджерах, публікуватися на тематичних ресурсах.

Кожен власник бізнесу повинен розуміти, що конкуренція зростає у геометричній прогресії і часу на зволікання вже немає. Навіть найкращий продукт сам себе продавати не буде. Ваші потенційні клієнти бажають отримати якомога більше інформації про товар, яким будуть користуватися. Саме просування сайту в Google є своєрідний початок до першого контакту з клієнтом.

Таким чином, під поняттям «комплексне просування сайту» мається на увазі не тільки постійне вдосконалення самого сайту, а й активне використання якомога більшої кількості доступних інструментів інтернет-маркетингу з метою отримання нових відвідувачів. До основних напрямів внутрішньої оптимізації відносяться такі [3]:

- виправлення технічних помилок. Вони можуть бути різними – дублі

сторінок, порожні сторінки та URL, які видають помилку 404, 301 та 302 редіректи, через неправильне використання яких сайт може значно довше грузитися або навіть потрапити під фільтри пошукових систем і багато іншого. Перевірка сайту фахівцями в рамках технічного аудиту, зазвичай, проводиться за кількома десятками параметрів;

- робота над юзабіліті. Сайт повинен бути зручним для користувачів, зі зрозумілою навігацією, щоб при переході на будь-яку його сторінку вони могли швидко розібратися, що до чого. Перевірити юзабіліті сайту можна декількома способами: Перший – дати сторонній людині і нехай спробує знайти усю необхідну інформацію. Другий спосіб – провести та проаналізувати поведінку вже існуючих відвідувачів сайту за допомогою карти кліків Plerdy. Використання другого методу значно скорочує час на аналіз дій користувачів, у порівнянні із першим способом, показує яка послідовність кліків була здійснена на сторінці, глибину прокрутки;

- створення контенту. Якісний, структурований і унікальний контент у сучасних умовах – необхідна умова для успішного розвитку будь-якого веб-проєкту. Також контент повинен створюватися на основі попередньо зібраного семантичного ядра, до якого входять ключові запити, важливі для цієї тематики бізнесу. Враховуючи потреби клієнтів у 2021 році контент повинен бути актуальним, чітко структурованим, цікавим, а головне відповідати на його питання чи вирішувати його наявні проблеми. Лише у такому випадку контент буде працювати Вам на руку;

- оптимізація метатегів. Ці теги – Title і Description – не відображаються на сторінці, але використовуються пошуковими системами для кращого розуміння її змісту. Їхня релевантність та оптимізація – дуже важливі складові внутрішніх робіт із SEO. Не варто забувати, що метатеги також варто писати на основі попередньо зібраного семантичного ядра, до якого входять ключові запити даної сторінки;

- параметр Alt для зображень. Технічно він теж відноситься до категорії метатегів і повинен містити текст, який описує зміст фотографії. У деяких

тематиках переходи на сайт із пошуку за зображеннями Google можуть генерувати відчутну частку трафіку;

- файли `sitemap.xml` і `robots.txt`. Ці файли містяться в кореновому каталозі сайту і мають свій спеціальний синтаксис. Файл `sitemap.xml` містить інформацію про всі важливі сторінки сайту, що прискорює їхнє індексування роботом, а файл `robots.txt` містить вказівки для пошукового робота щодо того, які сторінки потрібно відображати в пошуковому індексі, а які ні. Зверніть увагу, що для сайтів на яких присутні менше 1000 сторінок – карта сайту є не обов’язковою;

- швидкість завантаження. Google офіційно заявляє, що сайти, сторінки яких занадто повільно завантажуються, ранжуватимуться в пошуку нижче, ніж сайти, сторінки яких відкриваються швидко. Одним із важливих етапів комплексного просування сайту в Google є перевірка швидкості завантаження та її вдосконалення. Ідеальним результатом до якого варто прагнути – 1,5 сек.

Інструкція з розміщення сайту в інтернеті. Завдання для веб-сервера:

- організувати роботу декількох проектів на сервері, щоб користувачі, які ними займаються, не мали доступу до сусідніх проектів;
- обмежити можливість огляду файлової системи певної папкою для користувача проекту;
- обмежити можливість запуску бінарних файлів користувачами;
- обмежити можливість відкриття портів на сервері;
- автоматизувати додавання користувача в систему: створення папки, конфігураційного файлу `apache`, користувачів `mysql` або `postgres`.

На першому етапі потрібно оновити систему. Ставимо свіжі версії пакетів (Див. лістинг 2.10).

Лістинг 2.10 – Список репозиторіїв.

```
/etc/apt/sources.list
deb http://mirror.yandex.ua/debian/lenny main
deb http://mirror.yandex.ua/debian/lenny contrib non-free
deb-src http://mirror.yandex.ua/debian/lenny main
deb-src http://mirror.yandex.ua/debian/lenny contrib non-free
# Security fix
```

```
deb http://security.debian.org/lenny/updates main
deb http://security.debian.org/lenny/updates contrib non-free
deb-src http://security.debian.org/lenny/updates main
deb-src http://security.debian.org/lenny/updates contrib non-free
# apt-get update
# apt-get dist-upgrade
```

Генеруємо ключі для SSH. Щоб виключити можливість перехоплення паролльної фрази, сгенеруємо rsa ключі для входу на сервер (Див. лістинг 2.11).

Лістинг 2.11 – Генерация ключів для SSH

```
neoveneficus@book: ~ $ ssh-keygen
Generating public / private rsa key pair.
Enter file in which to save the key (/home/neoveneficus
/.ssh/id_rsa):
/home/neoveneficus/.ssh/id_rsa already exists.
Overwrite (y / n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/neoveneficus/.ssh/id_rsa.
Your public key has been saved in /home/neoveneficus
/.ssh/id_rsa.pub.
The key fingerprint is:
cb: 07: dd: 67:21:37: ab: 93: e1: 60:40: ce: 0E: b8: b8: e3
neoveneficus@book
```

Необхідно звернути увагу на наступне: ключі генеруються на робочій машині, а потім публічний ключ заливається на сервер (Див. лістинг 2.12).

Лістинг 2.12 – Генерация ключів на робочій машині

```
$ scp ~ /.ssh/id_rsa.pub root@217.212.252.146: .ssh
/authorized_keys
$ cat ~ /.ssh/id_rsa.pub | ssh root@217.212.252.146 "cat> ~
/.ssh/authorized_keys; chmod 600 ~ /.ssh/authorized_keys"
```

Тепер ми зможемо заходити на сервер – по ключу, з паролем або без – в залежності від налаштувань. Хочеться підкреслити, що якщо був згенерований ключ без паролю, то рівень безпеки сервера еквівалентний рівню безпеки секретного ключа на робочій машині.

Перейдемо до налаштування файлової системи. Права користувачів.

Для хостингу сайтів буде виділено окрему папку (Див. лістинг 2.13).

Лістинг 2.13 – Виділення окремої папки для хостингу сайту

```
# cd /
# mkdir-m 755 web
```

Домашні папки наших користувачів будуть такого вигляду (Див. лістинг 2.14).

Лістинг 2.14 – Домашні папки користувачів

```
/Web/site1
/Web/site2
```

Створюємо користувачів і розставляємо права (Див. лістинг 2.15).

Лістинг 2.15 – Створення користувачів

```
# useradd site1-b /web /-m-U-s /bin/false
# passwd site1
# chmod 754 /web/site1
# mkdir-p-m 754 /web/site1/public_html/www
# mkdir-p-m 777 /web/site1/tmp
# chmod + t /web/site1/tmp
# chown-R site1: site1 /web/site1/
```

Користувач site1 буде мати домашню теку /web/site1. Ключ -m означає, що папка буде створена автоматично. -U - так само буде створена однойменна група, куди користувач буде поміщений. Користувач не матиме шелу. Все, що буде доступно з веб - буде знаходитися в папці public_html. Якщо буде потреба в піддоменах, то їх буде розміщено в папках поряд з www.

Перейдемо до конфігурування веб-сервера Apache.

За замовчуванням web-сервер apache працює від одного користувача для всіх сайтів, що знаходяться на сервері. А це означає, що, скориставшись web-шелом, можна прочитати файли сусідніх проектів.

Для виправлення цієї проблеми нам знадобиться встановити модифіковану версію apache (Див. лістинг 2.16). Пакет називається apache2-mpm-itk. Встановивши пакет, ми отримаємо можливість в конфігураційному файлі

вказувати, від якого користувача і групи повинен працювати apache при обробці сайту.

Лістинг 2.16 – Модифікована версія apache

```
# apt-get install apache2-mpm-itk
```

Користувач і група задається наступним чином (Див. лістинг 2.17).

Лістинг 2.17 – Користувач, група

```
AssignUserId site1 site1
```

У той же час потрібно, щоб, використовуючи web-шелл (Див. лістинг 2.18), не можна було правити файли нашого проекту, крім тих, на яких стоять права o + w.

Лістинг 2.18 – Веб-шелл

```
drwxr-xr - 2 site1 site1 4096 5 березня 10:17.
drwxr-xr-x 3 site1 site1 4096 5 березня 10:14 ..
-Rwxr-x --- 1 site1 site1 0 5 березня 10:14 index.php
-Rwxrw ---- 1 site1 site1 0 5 березня 10:17 tmp.txt
```

Для цього в нашому конфігураційному файлі ми будемо писати:
AssignUserId www-data site1.

Таким чином, apache зможе прочитати index.php, виконати і віддати вбраузер, але не зможе змінити його, а tmp.txt змінити зможе.

Важливий момент – потрібно заборонити консоль у користувача www-data (Див. лістинг 2.19).

Лістинг 2.19 – Заборона консолі у користувача www-data

```
# usermod-s /bin/false www-data
```


Налаштуємо окремий tmp для кожного веб-сайту. Запобігаємо інклюд сесій з сусіднього сайту. Забороняємо php виходити вище користувацької домашньої директорії.

У конфігураційний файл сайту потрібно додати директиву `open_basedir`, `upload_tmp_dir`, `session.save_path` (Див. лістинг 2.20). Отримуємо такий файл конфігурації для веб-сайту.

Лістинг 2.20 – Додавання директиву `open_basedir`, `upload_tmp_dir`, `session.save_path`

```
/etc/apache2/sites-available/site1
<VirtualHost *:80>
DocumentRoot "/web/site1/public_html/www/"
ServerName "test1"
ErrorLog /web/site1/error_log
CustomLog /web/site1/access_log combined
AssignUserId www-data site1
php_admin_value open_basedir "/web/site1/::."
php_admin_value upload_tmp_dir "/web/site1/tmp"
php_admin_value session.save_path "/web/site1/tmp"
</VirtualHost>
```

Після створення файлу конфігурації потрібно зробити сайт активним за допомогою команди: `# a2ensite site1`. А потім перечитати файл конфігурації: `# /etc/init.d/apache2 reload`.

Налаштуємо Sendmail (Див. лістинг 2.21). Щоб php вмів відправляти листи, про робляємо наступні операції.

Лістинг 2.21 – Налаштування Sendmail

```
/etc/php5/apache2/php.ini [розкоментуємо і змінимо рядок];
+ + Sendmail_path = /usr/sbin/exim4-t
```

Для конфігурації Exim4 (за умовчанням як MailTransferAgent використовується саме він) існує пакет `exim4-config`. Скористатися ним можна так: `# dpkg-reconfigure exim4-config`.

Для доступу до каталогів веб-сервера необхідно налаштувати FTP-сервер (Див. лістинг 2.22).

Лістинг 2.22 – Налаштування FTP-сервер

```
vsftpd: # apt-get install vsftpd
```

Змінити файл конфігурації (Див. лістинг 2.23).

Лістинг 2.23 – Змінюємо файл конфігурації

```
/etc/vsftpd.conf listen = YES # Забороняємо анонімний доступ;
anonymous_enable = NO # Відкриваємо локальним користувачам доступ
до домашніх директорій по FTP :
local_enable = YES # Дозволяємо команди на запис

write_enable = YES
dirmessage_enable = YES
xferlog_enable = YES
connect_from_port_20 = YES
ascii_upload_enable = YES
ascii_download_enable = YES
ftpd_banner = Welcome to our FTP service.
# Обмежують користувачів в домашніх теках
chroot_local_user = YES
secure_chroot_dir = /var/run/vsftpd
pam_service_name = vsftpd
rsa_cert_file = /etc/ssl/certs/vsftpd.pem
Перезавантажуємо vsftpd:
# /etc/init.d/vsftpd reload
```

Перейдемо до налаштування MySQL: # apt-get install mysql-server
phpmyadmin.

Створюємо нового користувача (Див. лістинг 2.24).

Лістинг 2.24 – Створення нового користувача

```
# echo "CREATE USER 'site1' @ 'localhost' IDENTIFIED BY
'Пароль_для_пользователя_site1';
GRANT USAGE ON *.* TO 'site1' @ 'localhost' IDENTIFIED BY
'Пароль_для_користувача_site1' WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;
CREATE DATABASE IF NOT EXISTS site1 DEFAULT CHARACTER SET utf8
COLLATE utf8_unicode_ci;
GRANT ALL PRIVILEGES ON site1.* TO 'site1' @ 'localhost'; "
```

Перейдемо до налаштування між мережевого екрану iptables для захисту хостингу (Див. лістинг 2.25).

Лістинг 2.25 – Перехід до налаштування між мережевого екрану iptables

```
# apt-get install arno-iptables-firewall
Do you want to manage the firewall setup with debconf? :Так
External network interfaces: eth0
Open external TCP-ports: 21 22 53 80 443 3128 5432 5001 5900
6881: 6889
Open external UDP-ports: 53 3130 5001 6881:6889
Internal network interfaces: <в нашому випадку залишаємо порожнім>
Погоджуємося на перезавантаження файрвола.
Запуск Apache2 в chroot середовищі за допомогою libapache2-mod-
chroot.
```

Це створення спеціального середовища (пісочниці), ізольованої від оточення. Скажімо, процес apache не повинен мати доступу до папок home. У той же час в ізольовану середовищі потрібно брати все, що потрібно для роботи.

2.4 Висновки до другого розділу

Даний розділ містить постановку задачі на розробку, опис розробки логіки сайту (плагінів, темплейту, інших складових).

Веб-сайт передбачає два рівні доступу: адміністратор, користувач.

Звичайний користувач, тобто звичайний відвідувач зможе: переглядати сторінки сайту призначені для нього; реєструватись; авторизуватись для входу в особистий кабінет; переглядати інформацію у своєму профілі та змінювати її.

У користувача з правами адміністратора буде на порядок більше функціоналу включно з правами звичайного користувача, а саме у нього буде можливість керувати вмістом сайту, користувачами, їх правами. Також наведена інструкція з установки програмного забезпечення яка необхідна для роботи веб-сайту (СКБД, веб-сервер), тестування всіх складових сайту та обслуговування.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Шляхи підвищення життєдіяльності людини

Під життєдіяльністю розуміється властивість людини не просто діяти в життєвому середовищі, яке її оточує, а процес збалансованого існування та самореалізації індивіда, групи людей, суспільства і людства загалом в єдності їхніх життєвих потреб і можливостей [17].

Виробниче середовище забезпечує життєдіяльність під час виконання трудових обов'язків, в тому числі і працездатність, але при деяких обставинах, ці ж чинники можуть являти небезпеку і причиняти шкоду людині. Сукупність чинників трудового процесу і виробничого середовища, які впливають на здоров'я і працездатність людини під час виконання нею трудових обов'язків складають умови праці [17].

Опікуючись проблемою підвищення життєдіяльності людини та зростання продуктивності праці, сервіс з ремонту комп'ютерної техніки «ПК Майстер» не хтує значенням соціальних процесів, охорони здоров'я, забезпеченням гарантій в оплаті праці і соціальному захисті, забезпеченням безпеки праці, підвищення кваліфікації, постійного професійного навчання кадрів.

Ключовою проблемою при цьому є подолання низької мотивації до високопродуктивної праці, до розробки раціоналізаторських пропозицій.

Основними шляхами підвищення життєдіяльності людини та продуктивності є [12]:

- зміцнення матеріально-технічної бази (використання сучасних видів обладнання підвищить ефективність використання людських ресурсів);
- використання сучасних систем управління підприємством;
- створення методичної нормативно-правової бази підприємства допоможе чітко поставити перед працівниками завдання і окреслити рамки їх професійно-посадових обов'язків. За рахунок цього скоротиться час, що

витрачається на уточнення завдань, вирішення яких-небудь питань і суперечки щодо розподілу посадових обов'язків в рамках суміжних спеціалізацій;

- поліпшення організації праці, яке стосується як санітарно-гігієнічних умов, так і етичної складової організації робочих місць та місць відпочинку. Поліпшення умов праці, організація харчування в офісі та інше знизять втрати часу на вирішення невиробничих питань, а також підвищить лояльність співробітників до фірми;

- орієнтація соціальної політики на співробітника, мотиваційна складова – найактуальніший чинник, що впливає на зростання його продуктивності;

- підвищення лояльності співробітників до фірми, залучення їх до вирішення питань поточного та перспективного розвитку дасть серйозний поштовх до підвищення активності співробітника.

Слід відмітити особливо важливу роль в життєдіяльності людини природного освітлення, його ультрафіолетової частини спектру.

У сервісі з ремонту комп'ютерної техніки «ПК Майстер» в кабінетах, де є робочі місця співробітників, що працюють на комп'ютері, є наявні елементи природного та штучного освітлення. При цьому, на вікнах встановлено легко регульовані жалюзі, які дозволяють працівникам коригувати рівень освітлення в приміщенні. З метою досягнення максимального рівня безпеки і охорони праці при роботі з ПК, кабінети обладнані аптечками першої медичної допомоги, системами автоматичної пожежної сигналізації і вогнегасниками ВП-2.

Фізичні вправи в кінці дня допомагають контролювати стресові ситуації і зменшувати нервову збудливість.

Особливе місце серед засобів відновлення, які підвищують життєдіяльність людини, посідають раціональне харчування [20].

Важливим і необхідним є також постійне впровадження НТП у виробництво, стимулювання та мотивація працівників до праці, що є основним завданням власників підприємства. Отже, продуктивність праці на підприємстві є основним показником його успішності. Чим вища життєдіяльність людини та продуктивність праці, тим прибутковішою є бізнес-діяльність. Лише при

правильній організації, мотивації праці та впровадженні досягнень науки і техніки можна розраховувати на процвітання українських підприємств та економіки в цілому.

3.2 Економічне значення заходів щодо покращенню умов та охорони праці

Економічне значення охорони праці оцінюється за результатами, що отримані при зміні соціальних показників (поліпшення умов праці, підвищення безпеки праці, зниження виробничого травматизму) шляхом впровадження заходів з покращення умов праці: підвищення продуктивності праці; зниження непродуктивних витрат часу і праці; збільшення фонду робочого часу; зниження витрат, що пов'язані з плинністю кадрів через умови праці.

Закони про охорону праці передбачають положення, що стосуються економічних аспектів охорони праці. Наприклад, в Законі України „Про охорону праці” зазначено, що економічний прибуток підприємства залежить від фінансування заходів на охорону праці [1].

Особливості охорони праці при роботі з комп'ютером усервісі з ремонту комп'ютерної техніки «ПК Майстер».

Комп'ютерне обладнання підключається до електромережі лише за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників є спеціальні контакти для підключення нульового захисного провідника. Їх конструкція є такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні є зворотним.

Представимо в табличній формі (Див. Додаток А таблиця 1.1) виявлені небезпечні і шкідливі виробничі фактори відповідно до нормативних документів

із зазначенням обладнання або операцій, які є джерелом небезпечних і шкідливих виробничих факторів.

При роботі на комп'ютерах на працівників діють наступні шкідливі життю і здоров'ю психофізіологічні та фізичні виробничі фактори.

До фізичних факторів відносять [13]:

- підвищений рівень шуму (від вентилятора мікропроцесора та плат);
- високий рівень напруги в електричній мережі, торкання до якої може викликати нещасний випадок;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітного випромінювання;
- підвищений рівень напруженості електричного поля;
- несприятливе розподілення яскравості об'єктів в полі зору.

Наслідки регулярної роботи з комп'ютером без застосування захисних засобів можуть призвести до різноманітних захворювань, які негативно впливають як на самопочуття людини, так і на продуктивність праці. До наслідків регулярної роботи з комп'ютером без застосування захисних засобів відносять[16]:

- захворювання органів зору (60% користувачів);
- хвороби серцево-судинної системи (20%).

При роботі з ПК необхідно врахувати якість освітлення робочого місця. Основними вимогами до освітлення приміщень та робочих місць під час роботи з ПК є [26]:

- освітленість на робочому місці повинна відповідати характеру зорової роботи, який визначається трьома параметрами: об'єктом розрізнення – найменшим розміром об'єкта, що розглядається на моніторі ПК; фоном, який характеризується коефіцієнтом відбиття; контрастом об'єкта і фону;
- необхідно забезпечити достатньо рівномірне розподілення яскравості на робочій поверхні монітора, а також в межах навколишнього простору;
- на робочій поверхні повинні бути відсутні різкі тіні;

– в полі зору не повинно бути відблисків.

Особливої уваги заслуговують заходи дотримання протипожежної безпеки. Так, у всьому офісі лінії електромережі забезпечені від виникнення короткого замикання, а також від перепадів мережевої напруги, що може спричинити збої в роботі електронно-обчислювальної техніки.

Приміщення оснащені системою автоматичної пожежної сигналізації і вогнегасниками ВП-2 та ВВ-2. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легко займистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію.

З метою захисту від електромагнітного випромінювання доцільно замінити инаибільш поширені на підприємстві, але, вже морально застарілі монітори, які містять електронно-променеу трубку, на більше ергономічні рідкокристалічні, які характеризуються високою якістю відеовідтворення та сучасним дизайном [24].

Для захисту від електромагнітних випромінювань встановлені захисні фільтри для моніторів.

До економічних методів управління охороною праці відносяться [13]:

- утворення фонду охорони праці на виробництві;
- обов'язкове соціальне страхування працівників від нещасних випадків на виробництві і профзахворювань;
- збереження середньої заробітної плати працюючого в період простою у випадку відмови від дорученої роботи, якщо склалася виробнича ситуація, що представляє загрозу життю або здоров'ю працівників чи навколишньому природному середовищу.

Згідно законодавства України, роботодавець з метою досягнення економічного ефекту від упровадження заходів з охорони праці повинен забезпечити функціонування системи управління охороною праці, що є економічно вигідним для підприємства.

3.3 Проведення інструктажів з охорони праці

Інструктажі з питань охорони праці на підприємстві мають проходити усі працівники під час прийняття на роботу і в процесі трудової діяльності.

За характером і часом проведення розрізняють вступний, первинний, повторний, позаплановий та цільовий інструктажі [26].

Вступний інструктаж з охорони праці проводять: усім працівникам, яких беруть на постійну або тимчасову роботу.

Первинний інструктаж з охорони праці проводять з працівниками: новоприйнятими на постійну чи тимчасову роботу, відрядженими з іншого підприємства, з учнями, слухачами та студентами навчальних закладів.

Повторний інструктаж з охорони праці проводять не рідше: одного разу на три місяці – для робіт з підвищеною небезпекою, одного разу на шість місяців – для інших робіт.

Позаплановий інструктаж з охорони праці проводять у тому разі, якщо на підприємстві: при введенні в дію нових або змінених нормативних актів про охорону праці, змінено технологічний процес.

Цільовий інструктаж з охорони праці проводять у разі ліквідації аварії або стихійного лиха та проведення робіт, на які потрібен наряд-допуск, наказ або розпорядження.

Інструкція з охорони праці при роботі персональним комп'ютером [13].

Загальні положення:

1. За даною інструкцією користувач персонального комп'ютеру інструктується перед початком роботи (первинний інструктаж), а потім через кожні 6 місяців (повторний інструктаж);

2. До самостійної роботи на комп'ютерах допускаються особи, які:

- пройшли вступний інструктаж та інструктаж з питань охорони праці, пожежної безпеки при виконанні роботи на ПК з записом в журналах інструктажів;

- пройшли попередній та періодичний медичний огляди та за станом здоров'я допущені до виконання робіт на ПК;

- знають правила надання першої допомоги потерпілим;
- інструктаж і перевірку знань з електробезпеки.

3. Користувач повинен:

- не виконувати вказівок, які суперечать правилам охорони праці;
- пам'ятати про особисту відповідальність за виконання правил охорони праці та безпеку товаришів по роботі;
- вміти надавати першу медичну допомогу потерпілим від нещасних випадків;
- вміти користуватись первинними засобами пожежогасіння.

4. Основним обладнанням робочого місця користувача ПК є: ВДТ, системний блок, клавіатура, друкуючий пристрій (принтер), зовнішні пристрої пам'яті, сканер, "миша", блок безперервного живлення, робочий стіл, стілець та інше; допоміжним – шафи, полиці, сейф.

Загальні вимоги до обладнання робочого місця з ПК:

1) робоче місце для працюючих з відео терміналами необхідно необхідно розташувати таким чином, щоб до поля зору працюючого не потрапляли вікна, освітлювальні прилади, поверхні які мають властивість віддзеркалювання;

2) екран відео монітору ПК повинен знаходитись від очей користувача (надалі оператора) на відстані не менше 500 – 700 мм. Кут зору в межах 10-40 градусів. Найбільш раціональним є розташування екрану перпендикулярно до лінії зору оператора;

3) ПК повинен розташовуватися на відстані не ближче 1 метра від джерела тепла;

4) висота робочої поверхні стола повинна бути в межах 680-800 мм;

5) крісло повинно забезпечувати операторові зручні умови праці та фізіологічну раціональну робочу позу в процесі праці.

Для захисту оператора від електромагнітних випромінювань і електростатичних полів, які створює відео монітор необхідно використовувати захисні екрани.

Вимоги безпеки перед початком роботи:

- перед початком роботи працівник повинен зовнішнім оглядом перевірити цілісність корпусів системного блоку, відео монітора, принтера, клавіатури;

- перевірити цілісність кабелів живлення, місць їх підключення.

Вимоги безпеки під час роботи:

- не дозволяється використовувати рідинні або аерозольні засоби чистки поверхонь ПК;

- розташовувати принтер необхідно поряд з системним блоком так, щоб з'єднувальні шнури не були натягнуті.

Вимоги безпеки при закінченні роботи на ПК:

- закінчити і зберегти в пам'яті ПК файли, які знаходились у роботі;

- вимкнути принтер та інші пристрої, вимкнути системний блок.

Вимоги безпеки в аварійних ситуаціях:

- якщо після ввімкнення ПК відчувається запах горілого або при доторканні до металевих частин ПК відчувається дія електричного струму, потрібно негайно відключити ПК від електромережі та повідомити про це своєму керівникові;

- у випадку виникнення пожежі негайно розпочати гасіння наявними вогнегасниками (ВВ-2, ОП-2) і повідомити за телефоном 101.

Додатково, для збереження належного рівня здоров'я та професійної придатності працівників, виділили на підприємстві окреме побутове приміщення для перепочинку працівників і зняття ними нервово-емоційного напруження, що виникає при роботі з комп'ютером.

3.4 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи розглянуто питання Безпеки життєдіяльності, основи охорони праці на підприємстві. Опікуючись проблемою підвищення життєдіяльності людини та зростання продуктивності праці, сервіс з ремонту комп'ютерної техніки «ПК Майстер» не нехтує значенням соціальних процесів, охорони здоров'я, забезпеченням гарантій в оплаті праці і соціальному захисті, забезпеченням безпеки праці, підвищення кваліфікації, постійного професійного навчання кадрів.

Ключовою проблемою при цьому є подолання низької мотивації до високопродуктивної праці, до розробки раціоналізаторських пропозицій.

Основними шляхами підвищення життєдіяльності людини та продуктивності є: зміцнення матеріально-технічної бази, використання сучасних систем управління підприємством, поліпшення організації праці, яке стосується як санітарно-гігієнічних умов, так і етичної складової організації робочих місць та місць відпочинку.

Економічне значення охорони праці оцінюється за результатами, що отримані при зміні соціальних показників (поліпшення умов праці, підвищення безпеки праці, зниження виробничого травматизму) шляхом впровадження заходів з покращення умов праці.

ВИСНОВКИ

Результатом кваліфікаційної роботи є практично розроблений сайт для сервісу з ремонту комп'ютерної техніки «ПК Майстер». При розробці даного сайту було використано стек технологій та мов програмування, які дозволили зробити веб-сайт.

В першому розділі кваліфікаційної роботи:

- подано аналіз відомих рішень;
- проаналізовано односторінкові, багасторінкові веб-сайти та наведено порівняльну характеристику веб-сайтів сервісних центрів;

- розглянуто програмне забезпечення, мови програмування;
- здійснена постановка задачі, стадії та етапи розробки проєкту.

В другому розділі кваліфікаційної роботи:

- спроектовано структуру веб-сайту та структуру бази даних;
- розроблено шаблон (тему) та два плагіни;
- протестовано розробку веб-сайту на його валідність за допомогою: Chrome, Opera, InternetExplorer, MozillaFirefox, також було протестовано коректність роботи плагінів Catalog та Agent, для яких було встановлене спеціальне розширення User Agent Switcher;

- наведено інструкції щодо його обслуговування та наповнення веб сайту.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено такі питання:

- шляхи підвищення життєдіяльності людини;
- економічне значення заходів щодо покращенню умов та охорони праці;
- проведення інструктажів з охорони праці.

ПЕРЕЛІК ДЖЕРЕЛ

1. Закон України «Про охорону праці» від 14.10.1992 № 2694-XII із змінами та доповненнями від 04.02.2021 № 1213-IX // Відомості Верховної Ради України. 1992.
2. Анісімов А.В. Інформаційні системи та бази даних: навчальний посібник для студентів факультету комп'ютерних наук та кібернетики / А.В. Анісімов, П.П. Кулябко. – Київ: Основа, 2017. – 110 с.
3. Баженов В.А. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: підручник / В.А. Баженов, П.С. Венгерський, В.С. Гарвона. – Київ: Каравела, 2018. – 592 с.
4. Бережна О.Б. Інформатика та комп'ютерна техніка. 1 частина: навч. посібник / О.Б. Бережна. – Харків: ХНЕУ, 2018. – 164 с.
5. Бичков О.С. Основи сучасного програмування: підручник / О.С. Бичков. – Київ: Основа, 2017. – 285 с.
6. Вільна енциклопедія Вікіпедія [Електронний ресурс] // Режим доступу: <http://uk.wikipedia.org> 16.
7. Вінник В.Ю. Алгоритмічні мови та основи програмування: навчальний посібник / В.Ю. Вінник. – Житомир: ЖДТУ, 2018. – 328 с.
8. Вільна енциклопедія Вікіпедія [Електронний ресурс] // Режим доступу: <http://uk.wikipedia.org> 16.
9. Воронін А.М. Інформаційні системи прийняття рішень: навчальний посібник / А. М Воронін, А.С. Климова. – К.: НАУ-друк, 2017. – 136с.
10. Григорович В.Г. Методичні вказівки до виконання лабораторних і практичних завдань з курсу «Алгоритмізація та програмування». Частина 1: навч. посібник / В.Г. Григорович. – Дрогобич: Редакційно-видавничий відділ Дрогобицького державного педагогічного університету імені Івана Франка, 2018. – 540 с.
11. Глинський Я.М. Інформатика. Практикум з інформаційних технологій: навч. посібник / Я.М. Глинський. – Тернопіль: Підручники і посібники, 2015. –

304 с.

12. Грибан В.Г. Охорона праці: навч. посібник / В.Г. Грибан, О.В. Негодченко. – Київ: Центр учбової літератури, 2016. 280 с.

13. Геврик Є.О. Охорона праці: навч. посібн. / Є.О. Геврик. – Київ: Ніка-Центр, 2016. 280 с.

14. Годун В.М. Інформаційні системи і технології: навчальний посібник / В.М. Годун, Н.С. Орленко, М. А. Сендзюк. – Київ: КНЕУ, 2017. – 267 с.

15. Дибкова Л.М. Інформатика і комп'ютерна техніка: навч. посібник / Л.М. Дибкова. – Київ: Академвидавництво, 2016. – 463 с.

16. Жидецький В.Ц. Охорона праці користувачів комп'ютерів: навчальний посібник / В.Ц. Жидецький. – Львів: Афіша, 2017. 176 с.

17. Жернаков В. В. Трудове право: підручник / В.В. Жернаков, С.М. Прилипко, О.М. Ярошенко. – Харків: Право, 2015. 496 с.

18. Інформаційні технології: навчально-методичний посібник / [М.І. Жалдак, О.А. Хомік, І.В. Володько та ін.]. Київ: Основа, 2015. – 194 с.

19. Інформаційні системи в сучасному бізнесі: навчальний посібник / [В.С. Пономаренко, І.О. Золотарьова, Р.К. Бутова та ін.]. Харків: ХНЕУ, 2016. – 484 с.

20. Іванов Ю.Ф. Трудове право України: навч. посібник / Ю.Ф. Іванов, М.В. Іванова. – Київ: Алерта, 2020. 442 с.

21. Інформаційні системи і технології в економіці: навчальний посібник / [В.С. Пономаренко, Р.К., Бутова І.В. Журавльова та ін.]. – Київ: Академія, 2016. – 542с.

22. Пасічник О. Г. Основи веб-дизайну: підручник / О.Г. Пасічник. – Київ: Основа, 2017. – 336 с.

23. Ковалюк Т.В. Алгоритмізація та програмування: підручник / Т.В. Ковалюк. – Львів : Магнолія, 2016. – 400 с.

24. Керб Л.П. Основи охорони праці: навч.-метод. посібник для самост. вивч. дисц. / Л.П. Керб. – Київ: КНЕУ, 2015. 252 с.

25. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посібник / П.О. Кравець. – Львів: Видавництво Львівської політехніки, 2017. – 624 с.

26. Основи охорони праці: підручник / [К.Н. Ткачук, М.О. Халімовський, В.В. Зацарний та ін.]. – Київ: Основа, 2016. 448 с.

27. Пономарьова Г.Ф. Нові педагогічні технології: навчально-методичний посібник / Г.Ф. Пономарьова, О.О. Бабакіна, С.Б. Бєляєв. – Харків: ХНЕУ, 2015. – 282с.

28. Сайт Apache [Електронний ресурс] // Режим доступу: <http://httpd.apache.org/>

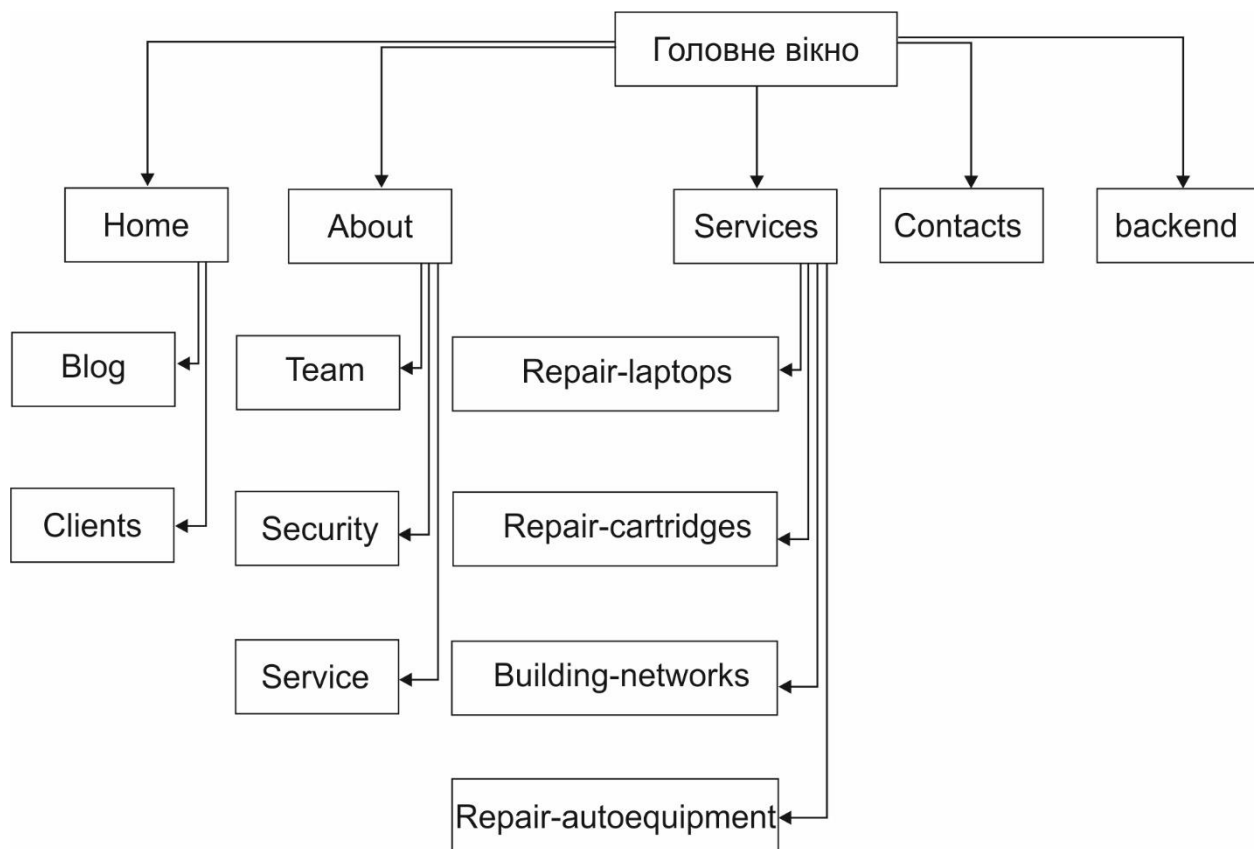
29. Сайт PHP [Електронний ресурс] // Режим доступу: <http://www.php.net/>

30. Сайт про Javascript [Електронний ресурс] // Режим доступу: <http://javascript/>

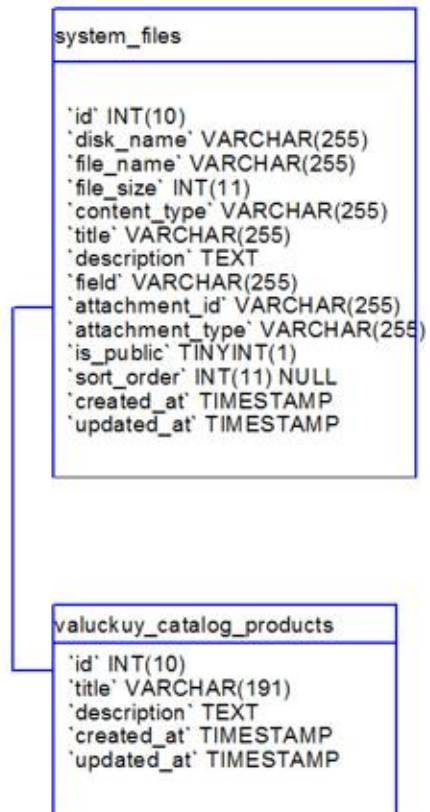
31. Шаховська Н.Б. Алгоритми та структури даних: підручник / Н.Б. Шаховська, Р.О. Голощук. – Львів : Магнолія, 2017. – 216 с.

ДОДАТКИ

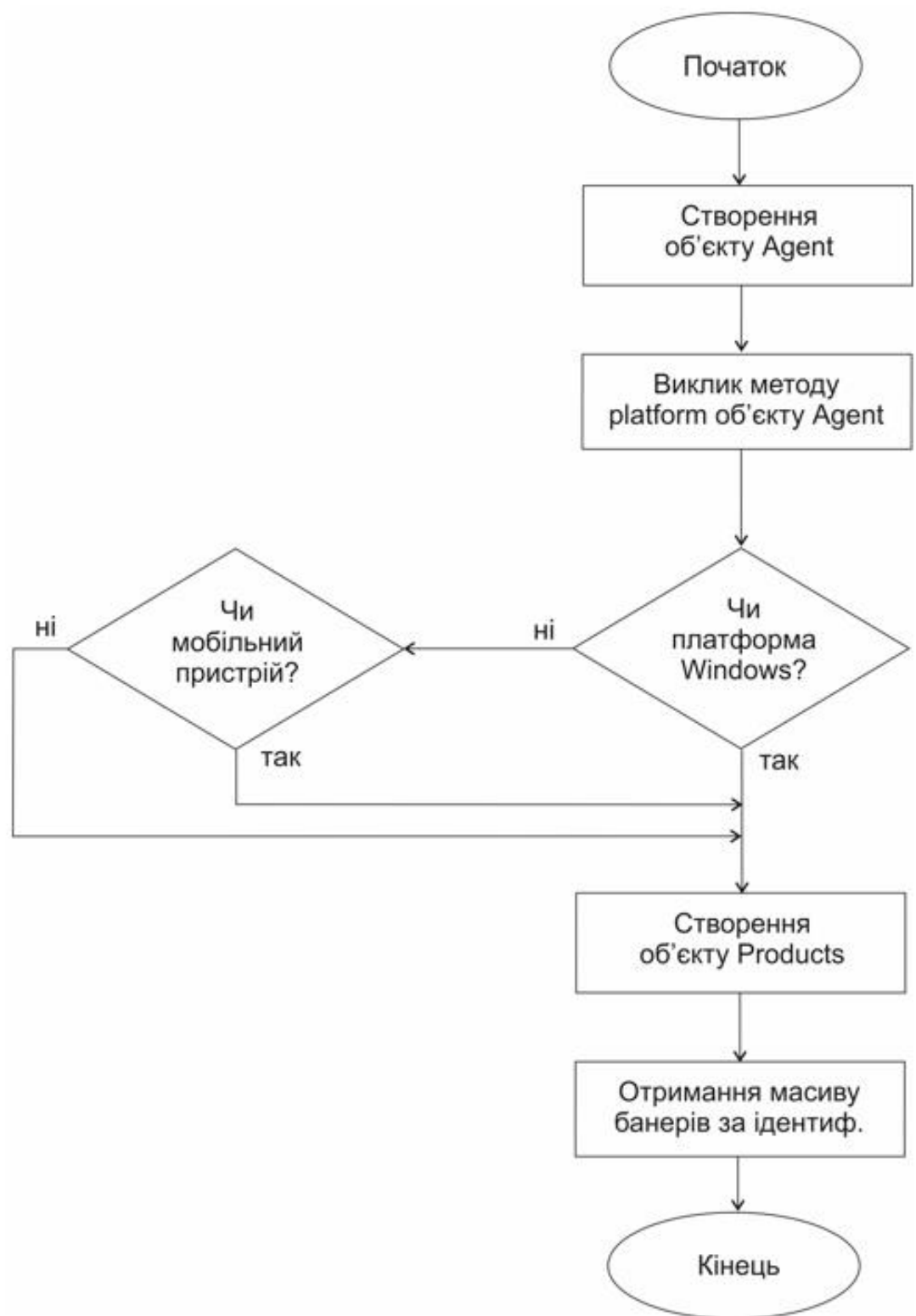
Структурна схема веб-сайту



База даних для плагіну Catalog



Блок-схема алгоритму



**Перелік небезпечних і шкідливих виробничих чинників на робочому місці
оператора ПЕОМ**

Назва небезпечних і шкідливих виробничих факторів	Джерело небезпечних і шкідливих виробничих факторів
Підвищена запиленість повітря робочої зони	Ксерокс, факс - частинки ультратонкого порошку (забруднює легені) Лазерний принтер - частинки тонера - подразнення дихальних шляхів
Підвищена або знижена температура повітря робочої зони;	Відсутність вентиляції, аерації, опалення
Підвищений рівень шуму на робочому місці	Стаціонарні телефони (відсутній захист від високих фонових і пікових акустичних навантажень),
Підвищена або знижена вологість повітря;	Неправильна робота кондиціонера, негерметичні вікна
Підвищена або знижена рухливість повітря;	Відсутність герметичності вікон
Підвищена або знижена іонізація повітря;	Наявність озону при використанні озонатора
Відсутність або нестача природного світла;	Відсутність отворів - бічних
Підвищена яскравість світла;	Використання матеріалів з коефіцієнтом відображення відмінним від нормованого значення

1	2
Знижена контрастність	Відсутність обліку фону і коефіцієнтів відбиття стін, підлоги
Підвищений рівень електромагнітного випромінювання	Монітор ПК, мобільні телефони
Перенапруження зорових аналізаторів	Екран монітора
Тривалі статичні навантаження	Мишка клавіатури - однотипні циклічні навантаження, при роботі з клавіатурою, незручне розміщення за столом

Лістинг коду Секція є власне шаблоном, на основі нього (Layout), фрагментів (Partials), вмістимого (Content) і буде сформовано кінцеву сторінку для користувача

```
<!DOCTYPE html>
<html>
<head>
    {% partial 'meta' %}
</head>
<body>
<div id="top-nav">
<div class="container">
</div>
</div>
<!-- Header -->
<header id="layout-header">
    {% partial "header" %}
<!-- Nav -->
<nav id="layout-nav">
    {% partial "nav" %}
</nav>
</header>
<!-- Content -->
<section id="layout-content">
<div class="container">
<div class="slider-wrapper title-style">
    {% partial "title"
        randomImage = randomImage %}
</div>
<div class="row module-homes">
<div class="col-sm-6 col-md-4">
<div class="module-home">
    {% page %}
</div>
</div>
<div class="col-sm-6 col-md-4">
<div class="module-home">
    {%
        placeholder
        news
title="Останні новини" %}
</div>
</div>
<div class="col-sm-6 col-md-4">
<div class="module-home">
    {%
        placeholder
        login
title="Авторизація" %}
</div>
{{ result }}
</div>
```



```

</div>
</div>
</section>
<!-- Slogan -->
<div id="layout-slogan">
<div class="container">
<div class="service-strip"></div>
</div>
</div>
<!-- Footer -->
<footer id="layout-footer">
        {% partial "footer" %}
</footer>
<!-- Scripts -->
        {% partial 'scripts' %}
<script src="{{ 'assets/javascript/home.js'|theme }}"></script>
</body>
</html>

```

Лістинг коду Скрипт валідації форми зворотнього зв'язку

```

$(window).on('ajaxInvalidField', function(event, fieldElement,
fieldName, errorMsg, isFirst) {
    var $field = $(fieldElement).closest('.form-group'),
        $help = $('<p />').addClass('help-block')
    if (!$field.length) {
        return
    }
    event.preventDefault()
    if (errorMsg) {
        $help.text(errorMsg.join(', '))
    }
    $help.addClass('form-field-error-label')
    $field.addClass('has-error')
    $(window).one('ajaxErrorMessage', function(event, message){
event.preventDefault()
    })
    if ($('.form-field-error-label', $field).length > 0)
        return
    $field.append($help)
    if (isFirst) {
        $('html, body').animate({ scrollTop: $field.offset().top },
500, function(){
            fieldElement.focus()
        })
    }
})
$(document).on('ajaxPromise', '[data-request]', function() {
    var $form = $(this).closest('form'),
        $help = $('.form-field-error-label', $form)

```

```

if (!$form.length || !$help.length)
    return
$help.remove()
$('.form-group.has-error', $form).removeClass('has-error')

```

Лістинг коду Реєстрація плагіну для файлу Plugin.php

```

<?php namespace Valuckuy\Catalog;
use System\Classes\PluginBase;
/**
 * catalog Plugin Information File
 */
class Plugin extends PluginBase
{
    /**
     * Returns information about this plugin.
     *
     * @return array
     */
    public function pluginDetails()
    {
        return [
            'name' => 'Catalog',
            'description' => 'The plugin is for creating categories of
products (ads) for different purposes',
            'author' => 'Valuckuy Vasyl',
            'icon' => 'icon-leaf'
        ];
    }
    public function registerComponents()
    {
        return ['Valuckuy\Catalog\Components\Products' =>
'Catalog'];
    }
    public function registerNavigation()
    {
        return [
            'catalog' => [
                'label' => 'Каталог',
                'url' =>
\Backend::url('valuckuy/catalog/products'),
                'icon' => 'icon-list-alt',
                'order' => 500,
                'sideMenu' => [
                    'products' => [
                        'label' => 'Реклама',
                        'icon' => 'icon-list-alit',
                        'url' =>
\Backend::url('valuckuy/catalog/products')
                    ]
                ]
            ]
        ];
    }
}

```

Лістинг файлу (Плагін Catalog)**«Valuckuy\ Catalog\ Components\ Products.php»**

```
<?php namespace Valuckuy\Catalog\Components;
use Cms\Classes\ComponentBase;
use Valuckuy\Catalog\Models\Product;
class Products extends ComponentBase
{
    public function componentDetails()
    {
        return [
            'name'          => 'Products (Advertisement) Component',
            'description' => 'That class manages ads that were added
by user'
        ];
    }
    public function define Properties()
    {
        return [];
    }
    public function get Images Array( int $ads)
    {
        $result = [];

        $array = Product::where('id', $ads)->first()->attachments;
        foreach ($array as $key => $value) {
            $result[$key] = $value->path;
        }
        return $result;
    }
}
```

Лістинг файлу (Плагін Catalog)**«Valuckuy\Catalog\Controllers\products\config_form.yaml»**

```
# Record name
name: Product
# Model Form Field configuration
form: $/valuckuy/catalog/models/product/fields.yaml
# Model Class name
modelClass: valuckuy\Catalog\Models\Product
# Default redirect location
defaultRedirect: valuckuy/catalog/products
```

```
# Create page
create:
  title: Create Product
  redirect: valuckuy/catalog/products/update/:id
  redirectClose: valuckuy/catalog/products
# Update page
update:
  title: Edit Product
  redirect: valuckuy/catalog/products
  redirectClose: valuckuy/catalog/products
# Preview page
preview:
  title: Preview Product
```

Лістинг файлу (Плагін Catalog)

«Valuckuy\Catalog\Controllers\Products.php»

```
<?php namespace Valuckuy\Catalog\Controllers;
use BackendMenu;
use Backend\Classes\Controller;

class Products extends Controller
{
    public $implement = [
        'Backend.Behaviors.FormController',
        'Backend.Behaviors.ListController'
    ];
    public $formConfig = 'config_form.yaml';
    public $listConfig = 'config_list.yaml';
    public function __construct()
    {
        parent::__construct();
        BackendMenu::setContext('Valuckuy.Catalog', 'catalog', 'products');
    }
}
```

Лістинг файлу (Плагін Catalog)

«Valuckuy\Catalog\Controllers\products\config_list.yaml»

```
# Model List Column configuration
list: $/valuckuy/catalog/models/product/columns.yaml
# Model Class name
modelClass: Valuckuy\Catalog\Models\Product
# List Title
title: Manage Products
# Link URL for each record
recordUrl: valuckuy/catalog/products/update/:id
# Message to display if the list is empty
noRecordsMessage: backend::lang.list.no_records
# Records to display per page
```

```

recordsPerPage: 20
# Displays the list column set up button
showSetup: true
# Displays the sorting link on each column
showSorting: true
# Default sorting column
# defaultSort:
#     column: created_at
#     direction: desc
# Display checkboxes next to each record
# showCheckboxes: true
# Toolbar widget configuration
toolbar:
    # Partial for toolbar buttons
    buttons: list_toolbar
    # Search widget configuration
    search:
        prompt: backend::lang.list.search_prompt

```

Лістинг файлу (Плагін Catalog)

«Valuckuy\Catalog\Controllers\products\create.htm»

```

<?php Block::put('breadcrumb') ?>
<ul>
<li><a href="<?= Backend::url('valuckuy/catalog/products')
?>">Products</a></li>
<li><?= e($this->pageTitle) ?></li>
</ul>
<?php Block::endPut() ?>
<?php if (!$this->fatalError): ?>
<?= Form::open(['class'=>'layout']) ?>
<div class="layout-row">
<?= $this->formRender() ?>
</div>
<div class="form-buttons">
<div class="loading-indicator-container">
<button
                                type="submit"
                                data-request="onSave"
                                data-hotkey="ctrl+s, cmd+s"
                                data-load-indicator="Creating Product..."
                                class="btn btn-primary">
Create
</button>
<button
                                type="button"
                                data-request="onSave"
                                data-request-data="close:1"
                                data-hotkey="ctrl+enter, cmd+enter"
                                data-load-indicator="Creating Ads..."

```

```

        class="btn btn-default">
            Create and Close
    </button>
    <span class="btn-text">
        or
    <a href="<?=
Backend::url('valuckuy/catalog/products') ?>">Cancel</a>
    </span>
</div>
</div>
<?= Form::close() ?>
<?php else: ?>
<p class="flash-message static error"><?= e($this->fatalError)
?></p>
<p><a href="<?= Backend::url('valuckuy/catalog/products') ?>"
class="btn btn-default">Return to products list</a></p>
<?php endif ?>

```

Лістинг файлу (Плагін Catalog)

«Valuckuy\Catalog\Controllers\products\preview.htm»

```

<?php Block::put('breadcrumb') ?>
<ul>
<li><a href="<?= Backend::url('valuckuy/catalog/products')
?>">Products</a></li>
<li><?= e($this->pageTitle) ?></li>
</ul>
<?php Block::endPut() ?>
<?php if (!$this->fatalError): ?>
<div class="form-preview">
<?= $this->formRenderPreview() ?>
</div>

<?php else: ?>
<p class="flash-message static error"><?= e($this->fatalError)
?></p>
<p><a href="<?= Backend::url('valuckuy/catalog/products') ?>"
class="btn btn-default">Return to products list</a></p>
<?php endif ?>

```

Лістинг файлу (Плагін Catalog)

«Valuckuy\Catalog\Controllers\products\update.htm»

```

<?php Block::put('breadcrumb') ?>
<ul>

```

```

<li><a      href="<?=      Backend::url('valuckuy/catalog/products')
?>">Products</a></li>
<li><?= e($this->pageTitle) ?></li>
</ul>
<?php Block::endPut() ?
<?php if (!$this->fatalError): ?>
<?= Form::open(['class'=>'layout']) ?>
<div class="layout-row">
<?= $this->formRender() ?>
</div>
<div class="form-buttons">
<div class="loading-indicator-container">
<button
                                type="submit"
                                data-request="onSave"
                                data-request-data="redirect:0"
                                data-hotkey="ctrl+s, cmd+s"
                                data-load-indicator="Saving Product..."
                                class="btn btn-primary">
<u>S</u>ave
</button>
<button
                                type="button"
                                data-request="onSave"
                                data-request-data="close:1"
                                data-hotkey="ctrl+enter, cmd+enter"
                                data-load-indicator="Saving Product..."
                                class="btn btn-default">
Save and Close
</button>
<button
                                type="button"
                                class="oc-icon-trash-o btn-icon danger pull-
right"
                                data-request="onDelete"
                                data-load-indicator="Deleting Product..."
                                data-request-confirm="Do you really want to
delete this product?">
</button>
<span class="btn-text">
                                or
                                <a      href="<?=
Backend::url('valuckuy/catalog/products') ?>">Cancel</a>
</span>
</div>
</div>
<?= Form::close() ?>
<?php else: ?>
<p class="flash-message static error"><?= e($this->fatalError)
?></p>
<p><a href="<?= Backend::url('valuckuy/catalog/products') ?>"
class="btn btn-default">Return to products list</a></p>

```

Лістинг файлу (Плагін Catalog) «Valuckuy\Catalog\Models\Product.php»

```
<?php namespace Valuckuy\Catalog\Models;
use Model;
/**
 * Product Model
 */
class Product extends Model
{
    /**
     * @var string The database table used by the model.
     */
    public $table = 'valuckuy_catalog_products';
    /**
     * @var array Guarded fields
     */
    protected $guarded = ['*'];
    /**
     * @var array Fillable fields
     */
    protected $fillable = [];
    public $hasOne = [];
    public $hasMany = [];
    public $belongsTo = [];
    public $belongsToMany = [];
    public $morphTo = [];
    public $morphOne = [];
    public $morphMany = [];
    public $attachOne = [];
    public $attachMany = ['attachments' => ['System\Models\File']];
}
```


Лістинг файлу (ПлагінAgent)
«Valuckuy\Agent\Classes\Agent.php»

```
<?php
namespace Valuckuy\Agent\Classes;
use BadMethodCallException;
use Jaybizzle\CrawlerDetect\CrawlerDetect;
use Mobile_Detect;
class Agent extends Mobile_Detect
{
    /**
     * Список Desktop пристроїв.
     * @var array
     */
    protected static $desktopDevices = [
        'Macintosh' => 'Macintosh',
    ];

    /**
     * Список додаткових операційних систем
     * @vararray
     */
    protected static $additionalOperatingSystems = [
        'Windows' => 'Windows',
        'Windows NT' => 'Windows NT',
        'OS X' => 'Mac OS X',
        'Debian' => 'Debian',
        'Ubuntu' => 'Ubuntu',
        'Macintosh' => 'PPC',
        'OpenBSD' => 'OpenBSD',
        'Linux' => 'Linux',
        'ChromeOS' => 'CrOS',
    ];

    /**
     * Список додаткових браузерів.
     * @vararray
     */
    protected static $additionalBrowsers = [
        'Opera Mini' => 'Opera Mini',
        'Opera' => 'Opera|OPR',
        'Edge' => 'Edge',
        'UCBrowser' => 'UCBrowser',
        'Vivaldi' => 'Vivaldi',
        'Chrome' => 'Chrome',
        'Firefox' => 'Firefox',
        'Safari' => 'Safari',
        'IE' => 'MSIE|IEMobile|MSIEMobile|Trident/[.0-9]+',
        'Netscape' => 'Netscape',
        'Mozilla' => 'Mozilla',
    ];
}
```

```

];
/**
 * Список додаткових властивостей.
 * @var array
 */
protected static $additionalProperties = [
    // Операційні системи
    'Windows' => 'Windows NT [VER]',
    'Windows NT' => 'Windows NT [VER]',
    'OS X' => 'OS X [VER]',
    'BlackBerryOS' => ['BlackBerry[\w]+/[VER]',
'BlackBerry.*Version/[VER]', 'Version/[VER]'],
    'AndroidOS' => 'Android [VER]',
    'ChromeOS' => 'CrOS x86_64 [VER]',

    // Браузери
    'Opera Mini' => 'Opera Mini/[VER]',
    'Opera' => ['OPR/[VER]', 'Opera Mini/[VER]',
'Version/[VER]', 'Opera [VER]'],
    'Netscape' => 'Netscape/[VER]',
    'Mozilla' => 'rv:[VER]',
    'IE' => ['IEMobile/[VER];', 'IEMobile [VER]', 'MSIE [VER];',
'rv:[VER]'],
    'Edge' => 'Edge/[VER]',
    'Vivaldi' => 'Vivaldi/[VER]',
];
protected static $crawlerDetect;
public static function getDetectionRulesExtended()
{
    static $rules;

    if (!$rules) {
        $rules = static::mergeRules(
static::$desktopDevices, // NEW
static::$phoneDevices,
static::$tabletDevices,
static::$operatingSystems,
static::$additionalOperatingSystems, // NEW
static::$browsers,
static::$additionalBrowsers, // NEW
static::$utilities
        );
    }
    return $rules;
}
public function getRules()
{
    if ($this->detectionType ===
static::DETECTION_TYPE_EXTENDED) {
        return static::getDetectionRulesExtended();
    }
    return static::getMobileDetectionRules();
}

```

```

}
    public function getCrawlerDetect()
    {
        if (static::$crawlerDetect === null) {
static::$crawlerDetect = new CrawlerDetect();
        }
        return static::$crawlerDetect;
    }
    public static function getBrowsers()
    {
        return static::mergeRules(
static::$additionalBrowsers,
static::$browsers
        );
    }
    public static function getOperatingSystems()
    {
        return static::mergeRules(
static::$operatingSystems,
static::$additionalOperatingSystems
        );
    }
    public static function getPlatforms()
    {
        return static::mergeRules(
static::$operatingSystems,
static::$additionalOperatingSystems
        );
    }
    public static function getDesktopDevices()
    {
        return static::$desktopDevices;
    }
    public static function getProperties()
    {
        return static::mergeRules(
static::$additionalProperties,
static::$properties
        );
    }
    public function languages($acceptLanguage = null)
    {
        if ($acceptLanguage === null) {
            $acceptLanguage = $this-
>getHttpHeader('HTTP_ACCEPT_LANGUAGE');
        }
        if (!$acceptLanguage) {
            return [];
        }
        $languages = [];
        foreach (explode(',', $acceptLanguage) as $piece) {
            $parts = explode('; ', $piece);

```

```

$language = strtolower($parts[0]);
    $priority = empty($parts[1]) ? 1. :
floatval(str_replace('q=', '', $parts[1]));
    $languages[$language] = $priority;
}
arsort($languages);
return array_keys($languages);
}
protected function findDetectionRulesAgainstUA(array $rules,
$userAgent = null)
{
    // Loop given rules
    foreach ($rules as $key => $regex) {
        if (empty($regex)) {
            continue;
        }
        if ($this->match($regex, $userAgent)) {
            return $key ?: reset($this->matchesArray);
        }
    }
    return false;
}
public function browser($userAgent = null)
{
    return $this->findDetectionRulesAgainstUA(static::getBrowsers(), $userAgent);
}

public function platform($userAgent = null)
{
    return $this->findDetectionRulesAgainstUA(static::getPlatforms(), $userAgent);
}

public function device($userAgent = null)
{
    $rules = static::mergeRules(
static::getDesktopDevices(),
static::getPhoneDevices(),
static::getTabletDevices(),
static::getUtilities()
    );
    return $this->findDetectionRulesAgainstUA($rules,
$userAgent);
}
public function isDesktop($userAgent = null, $httpHeaders =
null)
{
    return !$this->isMobile($userAgent, $httpHeaders) && !$this->
isTablet($userAgent, $httpHeaders) && !$this->isRobot($userAgent);
}

```

```

public function isPhone($userAgent = null, $httpHeaders = null)
{
    return $this->isMobile($userAgent, $httpHeaders) && !$this->isTablet($userAgent, $httpHeaders);
}

public function robot($userAgent = null)
{
    if ($this->getCrawlerDetect()->isCrawler($userAgent ? : $this->userAgent)) {
        return ucfirst($this->getCrawlerDetect()->getMatches());
    }
    return false;
}

public function isRobot($userAgent = null)
{
    return $this->getCrawlerDetect()->isCrawler($userAgent ? : $this->userAgent);
}

public function version($propertyName, $type = self::VERSION_TYPE_STRING)
{
    if (empty($propertyName)) {
        return false;
    }
    if ($type !== self::VERSION_TYPE_STRING && $type !== self::VERSION_TYPE_FLOAT) {
        $type = self::VERSION_TYPE_STRING;
    }
    $properties = self::getProperties();
    if (true === isset($properties[$propertyName])) {
        $properties[$propertyName] = (array) $properties[$propertyName];
        foreach ($properties[$propertyName] as $propertyMatchString) {
            if (is_array($propertyMatchString)) {
                $propertyMatchString = implode("|", $propertyMatchString);
            }
            $propertyPattern = str_replace('[VER]', self::VER, $propertyMatchString);
            preg_match(sprintf('%s#is', $propertyPattern), $this->userAgent, $match);
            if (false === empty($match[1])) {
                $version = ($type === self::VERSION_TYPE_FLOAT ? $this->prepareVersionNo($match[1]) : $match[1]);
                return $version;
            }
        }
    }
    return false;
}

```

```

}

protected static function mergeRules(...$all)
{
    $merged = [];
    foreach ($all as $rules) {
        foreach ($rules as $key => $value) {
            if (empty($merged[$key])) {
                $merged[$key] = $value;
            } elseif (is_array($merged[$key])) {
                $merged[$key][] = $value;
            } else {
                $merged[$key] .= '|' . $value;
            }
        }
    }
    return $merged;
}

public function __call($name, $arguments)
{
    if (strpos($name, 'is') !== 0) {
        throw new BadMethodCallException("No such method exists:
$name");
    }
    $this->setDetectionType(self::DETECTION_TYPE_EXTENDED);
    $key = substr($name, 2);
    return $this->matchUAAgainstKey($key);
}

public function getResult()
{
    return $this->browser();
}
}
}

```

Лістинг файлу (ПлагінAgent)

«Valuckuy\Agent\Components\Agent.php»

```

<?php namespace Valuckuy\Agent\Components;
use Cms\Classes\ComponentBase;
use Valuckuy\Agent\Classes\Agent as ValuckuyAgent;
class Agent extends ComponentBase
{
    private $agent;
    public function componentDetails()
    {
        return [
            'name'          => 'Agent Component',
            'description' => 'A simple wrapper plugin'
        ];
    }
}

```

```

public function init()
{
    $this->agent = new ValuckuyAgent();
}
public function __call($name, $arguments)
{
    if (method_exists($this->agent, $name)) {
        $refMethod = new
        \ReflectionMethod('Valuckuy\Agent\Classes\Agent', $name);
        return $refMethod->invokeArgs($this->agent,
        $arguments);
    } else {
        return $this->agent->__call($name, $arguments);
    }
}
}

```

Лістинг файлу (ПлагінAgent)

«Valuckuy\Agent\Tests\Unit\PluginTest.php»

```

<?php namespace Valuckuy\Agent\Tests\Unit;
use Agent;
use PluginTestCase;
class PluginTest extends PluginTestCase
{
    /**
     * Checks if Agent facade is available.
     */
    public function testAgentFacade()
    {
        Agent::setUserAgent('Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)
        AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94
        Safari/537.36');
        $this->assertSame('Chrome', Agent::browser());
    }
}

```

Лістинг файлу(ПлагінAgent)

«Valuckuy\Agent\Classes\AgentServiceProvider.php»

```

<?php
namespace Valuckuy\Agent\Classes;
use Illuminate\Support\ServiceProvider;
class AgentServiceProvider extends ServiceProvider
{
    protected $defer = true;
}

```

```

public function register()
{
    $this->app->singleton('agent', function ($app) {
        return new Agent($app['request']->server());
    });
    $this->app->alias('agent', Agent::class);
}

public function provides()
{
    return ['agent', Agent::class];
}
}

```

Лістинг файлу (ПлагінAgent)

«Valuckuy\Agent\Classes\Facades\Agent.php»

```

<?php
namespace Valuckuy\Agent\Classes\Facades;
use Illuminate\Support\Facades\Facade;
class Agent extends Facade
{
    protected static function getFacadeAccessor()
    {
        return 'agent';
    }
}

```