

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматики

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

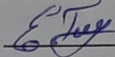
(повна назва кафедри (предметної, циклової комісії))

Бакалаврська дипломна робота на тему:

«WEB-ДОДАТОК ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ З ВИБОРУ
КОМП'ЮТЕРНИХ КОМПОНЕНТІВ»

Виконав: студент 4 курсу, групи ЗКН-206
спеціальності 122 – Комп'ютерні науки

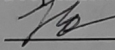
(цифр і назва напрямку підготовки, спеціальності)



Гуменчук Е. С.

(прізвище та інішіали)

Керівник: асистент кафедри КН



Малініч І. П.

(прізвище та інішіали)

« 07 » 06 2024 р.

Рецензент: к.т.н., доцент кафедри САІТ

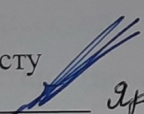


Крижановський Є. М.

(прізвище та інішіали)

« 07 » 06 2024 р.

Допущено до захисту

Зав. кафедри  Яровий А. А.

« 07 » 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А. А.
«07» 03 2024р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Гуменчуку Едуарду Сергійовичу

1. Тема роботи: Web-додаток для надання рекомендацій з вибору комп'ютерних компонентів
керівник роботи: Малініч І. П. асистент. каф. КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу «11» 03 2024 року №80
2. Термін подання студентом роботи: 27.05.2024 р.
3. Вихідні дані до роботи: не менше 20 компонентів кожного типу, налаштування та наповнення бази даних та додатку; час відгуку не повинен перевищувати 500 мілісекунд.
4. Зміст текстової частини (перелік питань, які потрібно розкрити): Аналіз та обґрунтування вибору CMS. Проектування частин веб-додатку. Реалізація частин веб-додатку. Розробка алгоритму функціонування частин веб-додатку. Тестування частин веб-додатку. Розробка інструкції користувача.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Загальні структурні схеми, алгоритм функціонування форми, скріншоти головного вікна, результати роботи веб-додатку.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	Малініч І. П., асистент каф. КН		

7. Дата видачі завдання 07.03.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Обґрунтування доцільності створення веб-додатку для надання рекомендацій з вибору комп'ютерних компонентів	06.05.24-11.05.24	
2	Аналіз існуючих рішень	11.05.24-20.05.24	
3	Проектування частин web-додатку	20.05.24-24.05.24	
4	Реалізація частин web-додатку	24.05.24-27.05.24	
5	Тестування частин веб-додатку	27.05.24-29.05.24	
6	Оформлення матеріалів до захисту БДР	29.05.24-06.06.24	

Студент


(підпис)

Гуменчук Е.С.
(прізвище та ініціали)

Керівник роботи


(підпис)

Малініч І.П.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 72 сторінок формату А4, на яких є 30 рисунків, список використаних джерел містить 22 найменування.

Бакалаврська дипломна робота присвячена створенню веб-додатку, що надає рекомендації щодо вибору компонентів комп'ютера. У цій бакалаврській роботі були розроблені форми та налаштування веб-додатку для надання рекомендацій щодо вибору комп'ютерних компонентів для систем керування контентом на базі WordPress з використанням мов програмування PHP та JavaScript, Cascading Style Sheets та Hypertext Markup Language.

Ключові слова: веб-додаток, веб-сайт, Wordpress, система управління вмістом, форми, підбір, комплектуючі, рекомендації.

ABSTRACT

The bachelor thesis consists of 72 pages of A4 format, on which there are 30 figures, the list of used sources contains 22 titles.

The bachelor's thesis is devoted to the creation of a web application that provides recommendations for the selection of computer components. This bachelor's thesis developed the forms and configuration of a web application to provide recommendations for the selection of computer components for WordPress-based content management systems using PHP and JavaScript programming languages, Cascading Style Sheets and Hypertext Markup Language.

Keywords: web application, website, Wordpress, content management system, forms, selection, components, recommendations.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОДАТКІВ З КОНФІГУРУВАННЯ	5
1.1 Обґрунтування доцільності	5
1.2 Огляд існуючих рішень	18
1.3 Постановка задачі дослідження	21
1.4 Висновок розділу	22
2 ПРОЕКТУВАННЯ ЧАСТИН WEB-ДОДАТКУ	23
2.1 Розробка структури веб-додатку	23
2.2 Розробка налаштувань та проектування панелей веб-додатку	26
2.3 Розробка алгоритмів функціонування частин веб-додатку	28
2.4 Висновок розділу	34
3 РОЗРОБКА ЧАСТИН WEB-ДОДАТКУ	36
3.1 Вибір інструментів розробки веб-додатку	36
3.2 Реалізація форми конфігуратора	43
3.3 Реалізація розділів веб-додатку	46
3.4 Тестування веб-додатку	49
3.5 Висновок розділу	55
ВИСНОВКИ	56
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	57
ДОДАТОК А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ	59
ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА WEB-ДОДАТКУ ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ З ВИБОРУ КОМП'ЮТЕРНИЙ КОМПОНЕНТІВ	60
ДОДАТОК В (обов'язковий) ЛІСТИНГ ПРОГРАМИ	61
ДОДАТОК Г (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	66

ВСТУП

Актуальність теми

Актуальність розробки веб-додатків для надання рекомендацій щодо вибору комп'ютерних компонентів. Розробка веб-додатків для надання рекомендацій щодо вибору комп'ютерних компонентів є дуже актуальною, оскільки існує багато користувачів, яким не вистачає інформації про сумісність компонентів. Часто саме цим користувачам потрібна допомога у виборі компонентів комп'ютера. Веб-додаток може бути найкращим способом надання такої інформації, оскільки дозволяє легко та зручно отримувати необхідні рекомендації.

Створення таких додатків також важливо з точки зору технологічного розвитку. Щороку з'являються нові технології та компоненти, які повинні враховувати певні критерії при складанні комп'ютера. Ці критерії змінюються з розвитком ринку, тому оновлювана база знань у вигляді веб-додатків - це процес вибору компонентів для користувача.

Під час бакалаврської роботи буде проведено аналіз найбільш популярних систем управління контентом (CMS), особливо для розробки веб-додатків, які надають рекомендації. Вибирається і тестується найкраща CMS для цієї мети. Крім того, аналізуватимуться різні типи структур веб-сайтів з метою вибору та розробки оптимальної структури веб-додатків. Важливим етапом роботи також буде розробка налаштувань і алгоритмів для функцій та тестування веб-додатку.

Дослідження допоможе створити ефективні інструменти, які полегшать користувачам вибір комп'ютерних комплектуючих з урахуванням сучасних технологічних вимог і змін на ринку.

Мета дослідження

Метою дослідження є розширення функціональних можливостей Web-додатку для надання рекомендацій з вибору комп'ютерних комплектуючих.

Задачі дослідження

Для досягнення поставлених цілей необхідно подальше вирішення наступних завдань дослідження:

- Огляд і аналіз відомих CMS;
- Розробка компонентів веб-додатків, що відповідають за форми та налаштування;
- Реалізувати компоненти веб-додатку у вибраній, відповідальній за форми, базу і налаштування;
- Перевірка роботи компонентів веб-додатку, що відповідають за форми, базу і налаштування.

Об'єктом дослідження є процес створення веб-додатку з надання рекомендацій.

Предметом дослідження є алгоритми та програмне забезпечення для створення складових частин веб-додатку орієнтованого на надання рекомендацій.

Практичною цінністю одержаних результатів є те, що розробники можуть використати багато функцій додатку для своїх розробок, таких як перевірка сумісності.

Апробація результатів роботи

Результати роботи були апробовані на LIII науково-технічній конференції підрозділів Вінницького національного технічного університету НТКП ВНТУ (2024).

Публікації

За результатами бакалаврської дипломної роботи опублікована теза доповіді на конференції «LIII Всеукраїнської науково-технічної конференції підрозділів ВНТУ (факультет інтелектуальних інформаційних технологій та автоматизації (2024))» (м. Вінниця, Україна) [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОДАТКІВ З КОНФІГУРУВАННЯ

1.1 Обґрунтування доцільності

Актуальність розробки веб-додатків для надання рекомендацій щодо вибору комп'ютерних компонентів. В останні роки велику популярність набули системи управління контентом (CMS). CMS (система управління контентом) - це програмне забезпечення, що використовується для організації веб-сайтів або інших інформаційних ресурсів в інтернеті або локальній мережі, така система використовується для управління, створення, редагування цифрового контенту та управління зображеннями, відео, документами, текстом та іншим мультимедійним контентом. Вона призначена для зберігання і публікації контенту.

Серед основних типів CMS можна виділити [2]:

1. Систему управління веб-контентом для управління веб-сайтами (онлайн-виданнями, блогами, форумами, корпоративними або особистими веб-сторінками).
2. Інтегровану CMS для управління корпоративним цдокументообігом.
3. Трансакційні CMS для електронної комерції.
4. Система, що забезпечує життєвий цикл документів (посібників, довідників, анотацій).
5. Цифрова бібліотека для управління цифровими активами (відео, графікою, презентаціями і т.д.).
6. Навчальна CMS для організації онлайн-курсів і тренінгів.
7. Платформа CMS, що підтримує автоматизацію роботи з комп'ютерними файлами, папками і програмами в певному програмному середовищі.

8. Корпоративна CMS для універсальної адаптації до потреб підприємства.

Компоненти CMS зазвичай складаються з 2 основних частин:

1. Додаток для управління вмістом (інтерфейс користувача), який дозволяє недосвідченим користувачам додавати, змінювати або видаляти вміст на сайті без втручання розробника.
2. Додаток для доставки вмісту, який керує доставкою та відображенням вмісту на веб-сторінках.

Типи установки CMS:

1. Локальна установка: можливість установки програмного забезпечення CMS на комп'ютер або сервер розробника забезпечує більшу гнучкість розробки, дозволяючи налаштовувати готові рішення CMS і створювати власні теми для вашого сайту. Прикладами таких CMS є WordPress, Drupal, Joomla і ModX.
2. Хмарна установка: CMS розміщується в середовищі провайдера, тому ви можете використовувати готові рішення без будь-яких змін. Це більш простий, але більш обмежений метод. Відомі приклади хмарних CMS включають WIX, SquareSpace та WordPress у хмарному режимі.

Ось основні критерії визначення функціональності CMS, які слід враховувати при виборі CMS:

1. Простота використання: наявність інтуїтивно зрозумілого інтерфейсу, доступність і простота використання.
2. Технологічність: використання найсучасніших технологій для підвищення якості, швидкості і надійності системи.
3. Програмна і апаратна Сумісність: Кросплатформеність, сумісність з різними СУБД, можливість підключення додаткових плагінів і модулів.
4. Розширюваність: можливість додавати нові функції та змінювати функціональність системи.

Серед ключових особливостей CMS можна виділити деякі, серед яких найбільш поширеними функціями CMS є:

1. Створення контенту (зазвичай виконується SMM-адміністратором або копірайтером).
2. Класифікація та індексація вмісту (за датою створення, автором, категорією, тегом та ключовим словом).
3. Сканування і перетворення контенту (текстового, графічного та інших форматів).
4. Збереження контенту в базі даних.
5. Тестування, перевірка достовірності та публікація контенту.
6. Підтримка, оновлення та управління змінами (моніторинг опублікованого контенту та реагування на необхідність внесення змін).

Переваги використання CMS:

1. Створення веб-сайту за допомогою системи управління контентом має багато переваг:
2. Вона проста у використанні і налаштуванні.
3. Можливість керувати контентом без навичок програмування.
4. Ви можете швидко редагувати вміст з адмін-панелі, не змінюючи код.
5. Регулярні оновлення та впровадження нових функцій.
6. Широкий вибір безкоштовних і платних плагінів і розширень.
7. Вбудована функція для SEO і просування сайту.
8. Підтримка розробників та спільноти.

У сучасному цифровому середовищі наявність веб-сайту є не просто бажанням, а необхідністю для будь-якого бізнесу чи організації. Створення та керування веб-сайтом може здатися складним завданням, але завдяки системам управління контентом (CMS) цей процес стає значно доступнішим. У цьому огляді буде проведено порівняльний аналіз трьох найпопулярніших CMS:

WordPress, Drupal та Joomla, аби допомогти вам у виборі оптимального рішення для ваших потреб.

WordPress – це CMS з відкритим вихідним кодом, яка здобула широку популярність завдяки простоті використання та гнучкості. Її простий інтерфейс та численні шаблони й плагіни роблять її доступною як для новачків, так і для досвідчених користувачів. WordPress чудово підходить для створення різноманітних веб-сайтів, від блогів та портфоліо до інтернет-магазинів та корпоративних сайтів.

WordPress – це система управління контентом з відкритим вихідним кодом, яка широко використовується для створення веб-сайтів завдяки простоті встановлення та використання. Її можна використовувати для всього, від блогів до складних веб-сайтів. Вбудована система тем і плагінів у поєднанні з чудовою архітектурою дозволяє створити практично будь-який веб-проект на основі WordPress; він використовує базу даних MySQL і написаний на мові програмування PHP. Вихідний код поширюється на умовах ліцензії GNU General Public License [3]. Офіційна сторінка веб-сайту WordPress показана на рисунку 1.1.

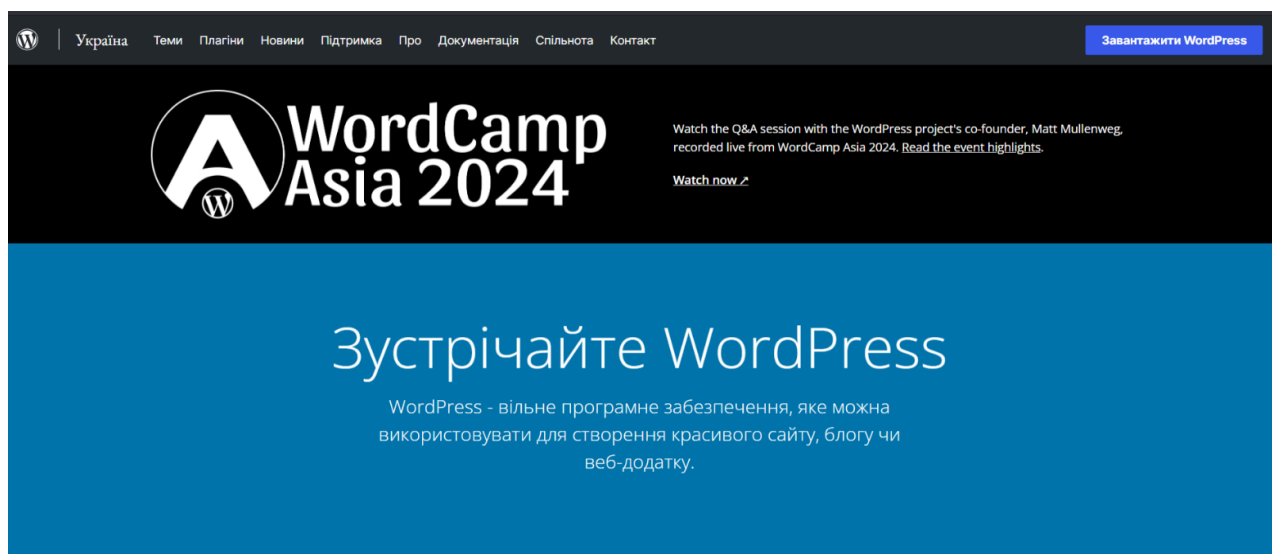


Рисунок 1.1 – Головна сторінка сайту Wordpress

Особливості платформи Wordpress включають

- Простий процес встановлення та налаштування.
- Відповідність веб-стандартам (XHTML, CSS).
- Плагіни, які забезпечують просту систему взаємодії з кодом, дозволяючи автоматичне встановлення та оновлення версій безпосередньо з адміністративної панелі.
- Підтримка тем, що дозволяє змінювати зовнішній вигляд і стиль відображення даних на сайті.
- Шаблони можна редагувати безпосередньо з адміністративної панелі.
- Готові теми та плагіни для спрощення розробки.
- Архітектурні можливості для комплексних рішень.
- SEO-оптимізація.
- Український переклад.
- Миттєва публікація.
- Підтримка RSS, Atom, Trackback та Pingback.
- Читабельні для людини URL-адреси.
- Редагування WYSIWYG.
- Багатомовне створення веб-сайтів.
- Публікація за розкладом.
- Керування правами доступу користувачів.
- Додавання файлів і зображень.
- Створення власного типу контенту з власною темою.
- Створення статичних сторінок.
- Категоризація та тегування.
- Коментування дописів.

Плагіни Wordpress — це елементи для розширення функціональності системи керуванням вмістом. У Wordpress можна встановлювати готові плагіни та створювати власні. Розробники Wordpress дали можливість користувачам створювати власні плагіни. Всі файли плагінів розміщуються в

теці wp-content/plugins. Його головний файл повинен бути написаний на мові PHP. Приклад елементу коду плагіна Wordpress зображений на рис. 1.2.

```
<?php
/*
Plugin Name: Wikipedia Link Plugin
Plugin URI: [[WordPress]]
Description:
Version: 0.0.1
Author: Wiki User
Author URI: http://uk.wikipedia.org
*/
function replace_footer_wiki(){
    echo("Блог зроблений на базі <a href=\"[[WordPress\"]>Wordpress]]
</a> ");
}
add_filter('get_footer', 'replace_footer_wiki');
?>
```

Рисунок 1.2 – Приклад коду Wordpress плагіна

Одне встановлення Wordpress дозволяє створювати та управляти лише одним блогом одночасно. Якщо є умова мати кілька блогів на одному сервері, можна встановити кілька екземплярів Wordpress у різні каталоги і створити окрему базу даних для кожного з блогів.

Гілка Wordpress Multi-User (Wordpress MU, або WPMU) надає можливість створювати кілька блогів в одній інсталяції. Wordpress MU також дозволяє кожному користувачу системи мати кілька блогів і керувати ними з одного панелі управління. Для кожного блогу створюється 8 таблиць в базі даних. [4].

Drupal - це ще одна CMS з відкритим вихідним кодом, відома своєю потужністю та масштабованістю. Її модульна структура та налаштовувані параметри роблять її ідеальним вибором для складних веб-сайтів та корпоративних порталів. Drupal потребує певних технічних знань для налаштування та обслуговування, але вона пропонує широкий спектр функцій та можливостей, недоступних у Wordpress.

Від особистих блогів до корпоративних і державних сайтів, Drupal служить основою для широкого спектру веб-сайтів, функціонуючи як серверна структура. Крім того, Drupal знаходить застосування в системах управління знаннями та полегшує бізнес-співпрацю [5].

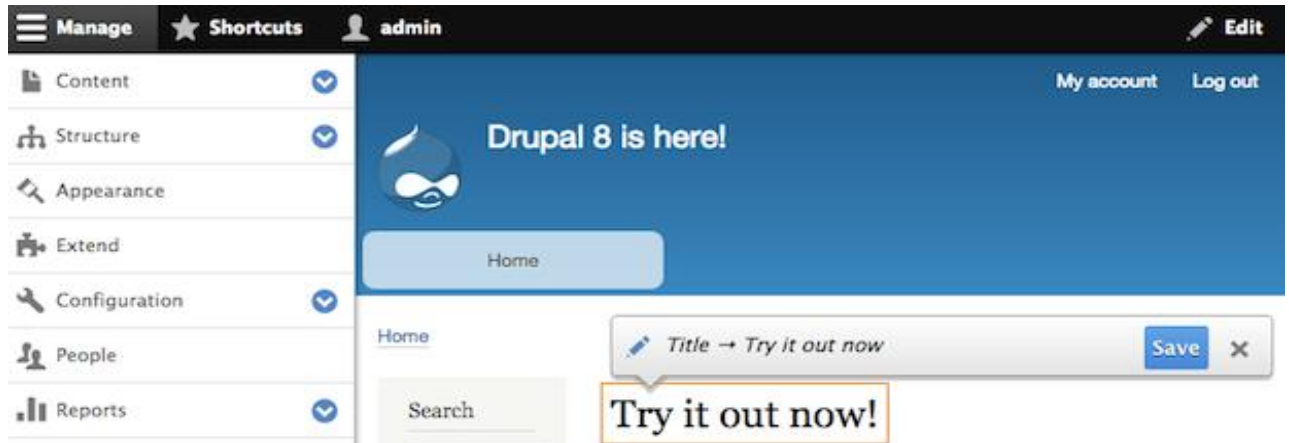


Рисунок 1.3 – Сторінка Drupal

CMS Drupal пропонує низку попередньо розроблених тем, а також можливість налаштувати власні теми. Ось основні функції CMS Drupal:

Платформа пропонує різні функції, такі як розширений пошук, можливість створювати блоги, форуми та книги, імпортувати новини з інших сайтів, писати коментарі, створювати та редагувати вміст, підтримувати декілька сайтів, створювати багатомовні сайти, запроваджувати обмеження контролю доступу на основі користувачів ролі та використання кешування та відкладеного завантаження вмісту.

Joomla - це CMS з відкритим вихідним кодом, яка пропонує баланс між простотою використання та функціональністю. Її інтерфейс є більш складним, ніж у WordPress, але вона все ж таки доступна для користувачів з базовими технічними знаннями. Joomla підходить для створення різноманітних веб-сайтів, включаючи блоги, веб-сайти візитки, інтернет-магазини та онлайн-спільноти. Для зберігання інформації використовуються бази даних MySQL, PostgreSQL або MS SQL [6]. На рис. 1.4 зображено головний сайт.

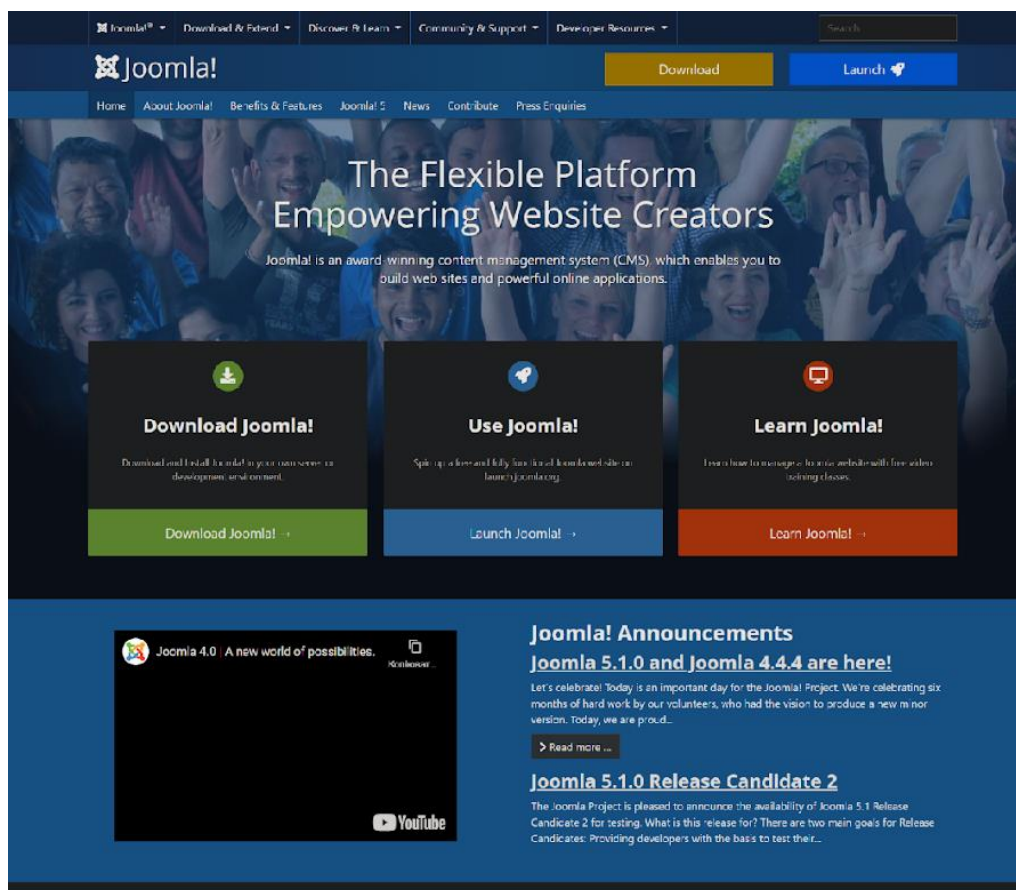


Рисунок 1.4 – Офіційна сторінка сайту Joomla!

Ця система має мінімальний набір інструментів під час початкової інсталяції та може бути розширена за потреби. Це зменшує навантаження на сервер і економить місце на хостингу [6].

Ключові характеристики Joomla!:

- Понад 8000 вбудованих модулів і компонентів, які можуть спростити розробку сайту.
- Можливість створювати власні модулі, плагіни, компоненти, шаблони та редагувати інтегровані.
- Багаторівневий доступ зареєстрованих користувачів до адміністративної та фронтальної частини сайту.
- Наявність менеджера з продажу новин.
- Підтримка створення багатомовних версій сторінок.

- Редагування матеріалів за допомогою візуального редактора TinyMCE (на кшталт текстових редакторів Word).
- Підтримка кешування сторінок.
- Система управління рекламними компаніями.

Всі три системи керування контентом (CMS), що розглядаються, є безкоштовними та розповсюджуються під ліцензією GNU General Public License (GPL) з відкритим кодом. Це означає, що користувачі можуть вільно завантажувати, використовувати, модифікувати та розповсюджувати ці системи. Відкритий код дозволяє розробникам переглядати та змінювати вихідний код для власних потреб, а також сприяти його вдосконаленню шляхом внесення своїх змін та оновлень. Така ліцензія сприяє прозорості, безпеці та розвитку програмного забезпечення завдяки активній участі спільноти.

Всі три CMS переважно написані на мові програмування PHP, що є однією з найпопулярніших мов для веб-розробки завдяки своїй простоті та широким можливостям. Для зберігання контенту та даних сайтів ці CMS використовують базу даних MySQL. MySQL є надійною та швидкою системою управління базами даних, яка дозволяє ефективно зберігати, організовувати та витягувати інформацію, необхідну для функціонування веб-сайтів. Поєднання PHP та MySQL забезпечує високу продуктивність та гнучкість у розробці веб-додатків.

Всі три CMS використовують теми та шаблони для контролю за зовнішнім виглядом сайтів. Це дозволяє користувачам легко змінювати дизайн та структуру своїх сайтів без необхідності глибоких знань у програмуванні. Теми та шаблони надають можливість застосовувати готові стилі та макети, що робить процес створення та налаштування сайту швидким та зручним. Користувачі можуть вибирати з широкого асортименту безкоштовних та платних тем, що задовольняють різні естетичні та функціональні потреби.

Для розширення функціональності сайтів всі три CMS використовують плагіни, модулі або розширення. Ці додаткові компоненти дозволяють легко

додавати нові можливості до сайтів, такі як інтеграція з соціальними мережами, інструменти для SEO-оптимізації, функції електронної комерції та багато іншого. Наявність великої кількості плагінів та модулів дозволяє користувачам налаштовувати свої сайти відповідно до специфічних вимог та цілей без необхідності писати власний код.

Кожна з цих CMS має велику та активну спільноту користувачів та розробників. Це спільнота надає підтримку новим користувачам, відповідаючи на питання та допомагаючи вирішувати проблеми. Також спільнота активно створює та підтримує додаткові теми, плагіни та модулі, що забезпечує постійний розвиток та покращення CMS. Участь спільноти дозволяє швидко виявляти та виправляти помилки, а також розробляти нові функції, що відповідають потребам користувачів.

1. Фокус та призначення:

- 1.1. WordPress: Найкраще підходить для блогів, простих веб-сайтів та сайтів малого бізнесу.
- 1.2. Drupal: Більш потужний та гнучкий, підходить для складних веб-сайтів, корпоративних порталів та веб-додатків.
- 1.3. Joomla: Займає проміжне місце між WordPress та Drupal, пропонуючи баланс простоти використання та функціональності.

2. Підхід до розробки:

- 2.1. WordPress: Має ядро з базовою функціональністю, яке можна розширити за допомогою плагінів.
- 2.2. Drupal: Більш модульна система, де кожен аспект функціональності сайту реалізовано окремим модулем.
- 2.3. Joomla: Використовує компонентну модель, яка поєднує в собі елементи WordPress та Drupal.

3. Крива навчання:

- 3.1. WordPress: Найпростіший у вивченні та використанні з трьох.

3.2. Drupal: Має більш круту криву навчання через свою модульну архітектуру.

3.3. Joomla: Трохи складніший за WordPress, але простіший за Drupal.

Вибір CMS для веб-додатку з рекомендаціями щодо комп'ютерних комплектуючих. Зважаючи на потреби, для веб-додатку з рекомендаціями щодо комп'ютерних комплектуючих краще використовувати WordPress або Joomla.

WordPress:

- Легкий у використанні та налаштуванні, що робить його ідеальним для швидкого запуску вашого веб-додатку.
- Велика кількість плагінів та тем, які можуть допомогти вам додати необхідні функції, такі як форми, логіка блогу та SEO.
- Велика та активна спільнота для отримання підтримки.

Joomla:

- Більш гнучкий, ніж WordPress, що дає вам більше можливостей для налаштування вашого веб-додатку.
- Легший у навчанні, ніж Drupal, завдяки більш дружньому інтерфейсу та документації.
- Велика кількість розширень для додавання необхідних функцій.

Drupal може бути кращим вибором, якщо потрібен веб-додаток з дуже складною логікою або високими вимогами до безпеки.

Для початківців та не професійних користувачів WordPress або Joomla можуть бути кращим вибором завдяки їх простоті використання. Якщо є хороший досвід роботи з PHP та веб-розробкою, Drupal може дати вам більше можливостей.

Всі три CMS є безкоштовними, але вам може знадобитися найняти розробника для створення вашого веб-додатку або налаштування його

відповідно до ваших потреб. Якщо ви плануєте розширювати свій веб-додаток у майбутньому, вам слід вибрати CMS, яка є гнучкою та масштабованою.

Для потреб веб-додатку з надання рекомендацій з вибору комп'ютерних комплектуючих підходить Wordpress через велику кількість переваг, зокрема:

- Повністю безкоштовний для встановлення, з можливістю багатьох модифікацій, його можна налаштувати та покращити відповідно до ваших потреб.
- Wordpress має низькі системні вимоги і працює з будь-яким хостингом.
- Простота встановлення.
- Системні файли будуть встановлені на вашому хостингу протягом декількох хвилин.
- Файли CMS легкі та економлять місце під час розміщення.
- Багато постачальників пропонують свої послуги з уже встановленим WordPress.
- Прозорий візуальний редактор.
- Контент додається за допомогою редактора WYSIWYG, який нагадує інтерфейс програми Office Word, тому навіть люди без досвіду програмування можуть працювати з контентом сайту.
- SEO оптимізація.
- Під час пошуку пошукові системи спрямовують вас на веб-сайти, створені за допомогою WordPress.
- Для повної оптимізації вам потрібно встановити плагін, який вводить метайнформацію, таку як заголовок, опис і ключові слова.
- Панель адміністратора та інтерфейс інтуїтивно зрозумілі і дозволяють будь-якому користувачеві керувати сайтом.
- Для цього не потрібні спеціальні знання.

- Користувачі, які знають код, мають можливість створювати та вдосконалювати теми самостійно, а також створювати та змінювати плагіни.
- В офіційному каталозі плагінів є понад 59 000 плагінів WordPress, які дуже корисні під час розробки та додавання нових функцій на ваш веб-сайт.

WordPress є основою для 43,2% усіх веб-сайтів у світі, що свідчить про його значну популярність серед розробників та користувачів. Це підтверджує його конкурентоспроможність і здатність до постійного розвитку, що приваблює нових фахівців [7].

Після аналізу найпопулярніших CMS для створення веб-додатку було обрано CMS WordPress. Вона перевершує інші системи за простотою використання, великою глобальною спільнотою, кількістю плагінів і тем. Вважається, що контент-менеджери, які будуть працювати з контентом у майбутньому, знайдуть інтерфейс WordPress набагато зручнішим, ніж у Joomla! чи Drupal [8].

Основними функціональними вимогами є:

1. Правильна робота конфігуратора: Конфігуратор повинен забезпечувати перевірку сумісності всіх характеристик компонентів. Він має виводити повідомлення про сумісність усіх компонентів між собою, а у випадку неправильного вибору користувача – сповіщати про проблеми з сумісністю та вказувати між якими компонентами виникає проблема.
2. Перевірка працездатності кнопок: Кожна кнопка повинна працювати та виконувати відповідну дію, яка зазначена на кнопці або біля неї.
3. Навігація: Всі навігаційні елементи повинні бути інтерактивними та відповідати принципам UI/UX дизайну, забезпечуючи зручну та інтуїтивну навігацію по сайту

Ці функціональні вимоги забезпечують основні аспекти роботи веб-додатку та сприяють створенню позитивного користувацького досвіду, а також гарантують ефективність та надійність веб-додатку

1.2 Огляд існуючих рішень

У цьому підрозділі буде детально розглянуто й проаналізовано існуючі додатки, що надають рекомендації з вибору комп'ютерних компонентів. Основною метою аналізу є визначення переваг і недоліків кожного з додатків, а також можливостей і перспектив для розробки власного програмного продукту. Це дозволить зрозуміти, які функції є найбільш корисними для користувачів, і які аспекти варто покращити у новому додатку.

Додаток «Telemart.ua» є одним з популярних рішень на ринку для конфігурування комп'ютерних систем. На головній частині екрану користувачеві пропонується широкий спектр функцій, включаючи конфігуратор, можливості порівняння товарів, кошик для покупок, вибрані товари та особистий кабінет. Цей додаток надає свої послуги за гроші, що свідчить про акцент на комерційні функції.

До переваг описаного додатка можна віднести зрозумілий та інтуїтивний інтерфейс, широкий спектр функціоналу, налаштування особистої персоналізації та в деяких моментах зручність у замовленнях, адже можна зібрати і купити всі комплектуючі зразу в додатку. З мінусів, серед яких один з найбільших - громіздкість додатку, адже він позиціонує себе саме як конфігуратор комп'ютера, але велика частина сайту - магазин з продажу різноманітної техніки. Також до мінусів можна віднести рекламу, адже вона також присутня на цьому сайті у вигляді «саморекламування» Головна сторінка додатку зображена на рисунку 1.5.

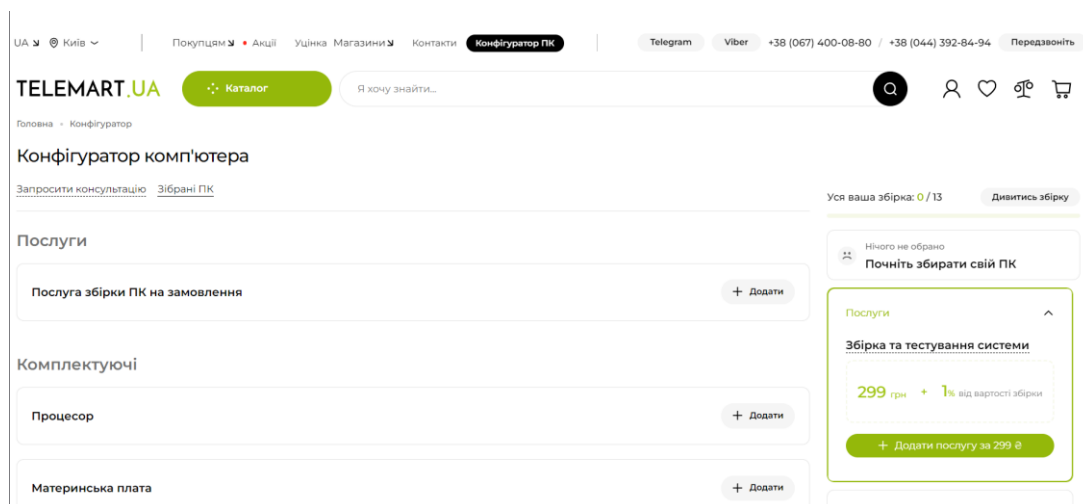


Рисунок 1.5 – Головна сторінка додатку «Telemart.ua»

Додаток «Elmir.ua» також позиціонує себе як конфігуратор ПК і має схожий функціонал з додатком «Telemart.ua». Він пропонує можливість складання комп'ютера та просування своїх товарів. Основною відмінністю є дизайн, але загальні переваги та недоліки залишаються аналогічними. Головне вікно додатку зображене на рисунку 1.6.

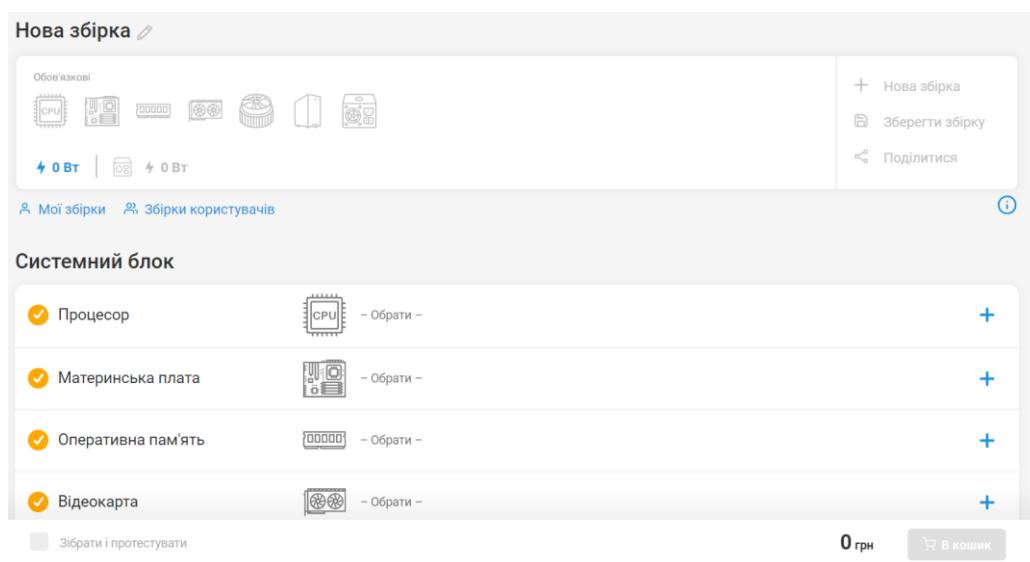


Рисунок 1.6 – Головна сторінка додатку «Elmir.ua»

Після ретельного аналізу головного додатка, потрібно зазначити, що перевагою в користуванні такого застосунка є приємна гама кольорів, сучасний дизайн та великий функціонал.

Додаток «ChipChip.ua» містить навігаційне меню, а також короткий опис функціоналу додатка. До переваг додатка можна віднести зручний інтерфейс, легке конфігурування, точне відстеження сумісності та велику базу товарів, сам додаток сильно нагадує попередні, має такі ж плюси та мінуси, але набагато більше пропонує ще свої послуг. Головна сторінка додатку зображена на рисунку 1.7.

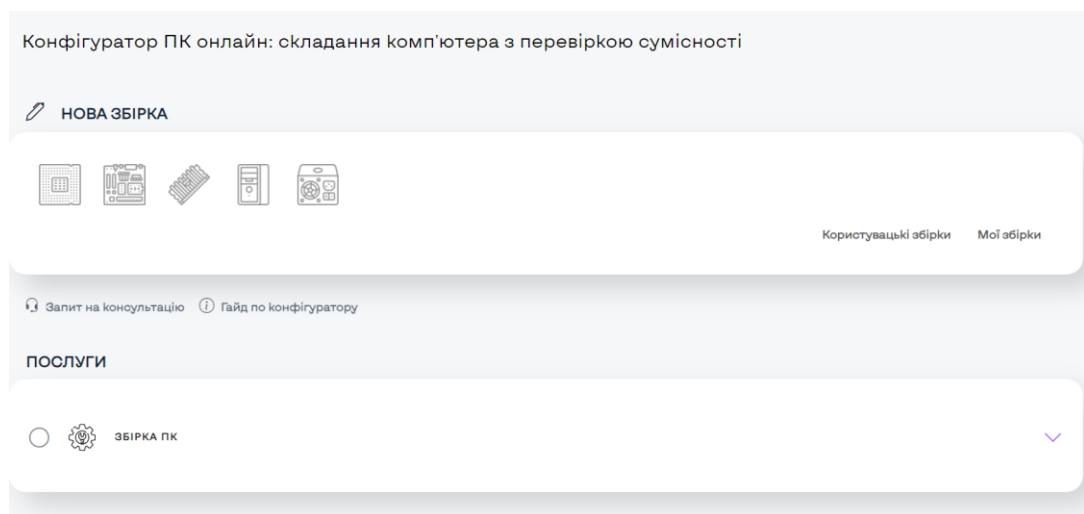


Рисунок 1.6 – Головна сторінка додатку «ChipChip.ua»

На основі проведеного аналізу існуючих додатків для конфігурування ПК було визначено, що створення WEB-додатку для надання рекомендацій з вибору комп'ютерних комплектуючих є доцільним та має практичне значення.

Такий WEB-додаток буде направлений саме з надання рекомендацій та перевірку сумісності, а не продажу товарів та просуванні платних послуг, а також корисні блоги для того, щоб користувач розумів, на яку інформацію потрібно акцентувати увагу. Такий WEB-додаток буде безкоштовним. Ключовими особливостями створення WEB-сторінки є налаштування інтерфейсу WEB-додатку, приділення уваги підбору кольорових відтінків. Налаштування інтерфейсу буде виконано у гарному дизайні та вибором гармонійних кольорів WEB-сайту. Порівняльна характеристика подана у таблиці 1.1.

Таблиця 1.1 - порівняльна характеристика WEB-додатків

Назва	Telemart	Elmir	ChipChip	Покращений додаток
Конфігуратор	Так	Так	Так	Так
Універсальність	Так	Так	Так	Так
Доступність	Так	Так	Так	Так
Відсутність просування платних послуг	Ні	Ні	Ні	Так
Відсутність реклами	Ні	Ні	Ні	Так
Відсутність перевантаження побічними функціями	Ні	Ні	Ні	Так
Безкорислива основа	Ні	Ні	Ні	Так

Особливостями розробки стануть наявність стильного, унікального дизайну та спеціального конфігуратора, що допоможе визначити сумісність компонентів та не зробити помилок в самостійній збірці.

1.3 Постановка задачі дослідження

Мета дослідження – розширення функціональних можливостей Web-додатку для надання рекомендацій з вибору комп'ютерних комплектуючих.

Задачі дослідження

Для досягнення поставлених цілей необхідно подальше вирішення наступних завдань дослідження:

- Огляд і аналіз відомих CMS;
- Розробка компонентів веб-додатків, що відповідають за форми та налаштування;
- Реалізувати компоненти веб-додатку у вибраній, відповідальній за форми, базу і налаштування;
- Перевірка роботи компонентів веб-додатку, що відповідають за форми, базу і налаштування.

Об'єктом дослідження є процес створення веб-додатку з надання рекомендацій.

Предметом дослідження є алгоритми та програмне забезпечення для створення складових частин веб-додатку орієнтованого на надання рекомендацій.

Практичною цінністю одержаних результатів є те, що розробники можуть використати багато функцій додатку для своїх розробок, таких як перевірка сумісності.

Апробація результатів роботи

Результати роботи були апробовані на LIII науково-технічній конференції підрозділів Вінницького національного технічного університету НТКП ВНТУ (2024).

Публікації

За результатами бакалаврської дипломної роботи опублікована теза доповіді на конференції «LIII Всеукраїнської науково-технічної конференції підрозділів ВНТУ (факультет інтелектуальних інформаційних технологій та автоматизації (2024))» (м. Вінниця, Україна) [1].

1.4 Висновок розділу

У цьому розділі описано, як створити веб-сайт за допомогою CMS, і перераховано переваги цього підходу. Проаналізовано найпопулярніші CMS системи та показано їх сильні та слабкі сторони. Крім того, для поставленого завдання розробки веб-додатку для надання рекомендацій з підбору комп'ютерних комплектуючих було обрано систему управління контентом. Поставлено функціональні вимоги на основі яких буде проводитися тестування додатку, а саме: правильна робота конфігуратора, повідомлення про сумісність компонентів та правильна робота кнопок.

2 ПРОЕКТУВАННЯ ЧАСТИН WEB-ДОДАТКУ

2.1 Розробка структури веб-додатку

Перш ніж почати працювати з веб-сайтом або будь-яким веб-ресурсом, спочатку потрібно чітко продумати його структуру. Це впливає на те, як веб-сайти відображаються та класифікуються в пошукових системах, а також на те, як користувачі сприймають їх вміст.

Структура веб-сайту - це розташування його сторінок, категорій, підкатегорій, тегів, товарів і т.д. можна сказати, що структура сайту простежує логічні зв'язки між сторінками. З технічної точки зору навігація по сайту - це набір посилань, розташованих у певному порядку, та набір під папок та файлів для веб-сайту.

Структура веб-сайту ділиться на зовнішню і внутрішню. Зовнішня структура показує макет сторінки і розташування блоків на ній, але можна сказати, що це більш розгорнуте відображення DOM-дерева веб-сайту. Внутрішня структура показує сторінки веб-сайту та сторінки, які належать до категорії. Саме це буде розглянуто в цьому підрозділі[9].

Типи структури веб-сайту:

- лінійна структура;
- деревовидна структура;
- сітчаста структура;
- змішана структура.

Коли створювати веб-сайт, потрібно вибрати структуру, яка відповідає цій вимозі, тому необхідно детальніше розглянути основні типи структур веб-сайтів.

На веб-сайті з лінійною структурою інформація представлена лінійним чином. Для кращого розуміння можна провести аналогію з процесом перегортання книги. Дана структура являє собою своєрідну ланцюжок, що складається з взаємопов'язаних компонентів. Найчастіше така структура

використовується для створення сайтів-візиток, основна мета яких - дати користувачеві зрозуміти контент, який викладений в певному порядку, оскільки лінійна структура не виділяє розділи, категорії і підкатегорії, і користувач може переміщатися по сторінці тільки за певним маршрутом. Наочно це видно на схемі лінійної структури сайту, що показана на рисунку 2.1.

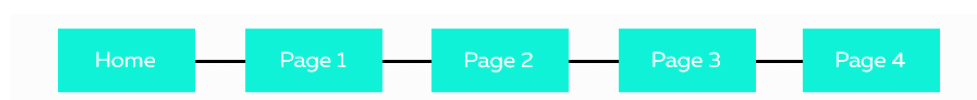


Рисунок 2.1 – Приклад схеми лінійної структури сайту

Деревоподібна структура сайту є найбільш оптимальною і підходить для більшості веб-ресурсів, таких як корпоративні сайти, сайти-візитки, каталоги, блоги, інтернет-магазини і портали. Плюсом такої структури є універсальність і миттєвий доступ до будь-якої сторінки, категорії або розділу веб-ресурсів [10]. У цій структурі є головна сторінка, з якої переходять посилання на інші сторінки і категорії, пости, картки товарів і тому подібне. Приклад деревовидної структури веб-сайту показаний на малюнку 2.2.

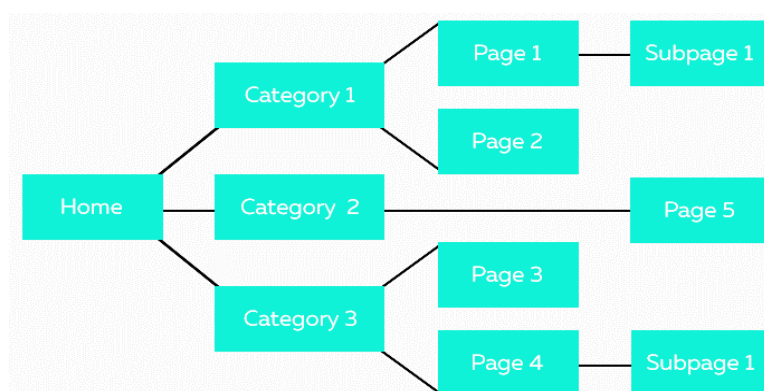


Рисунок 2.2 – Приклад схеми деревовидної структури сайту

У сітчастій структурі всі компоненти веб-сайту розміщені в окремих розділах. Якщо між інформаційними статтями є зв'язок, рекомендується використовувати таку структуру в каталозі. Використання структури такого

типу може заплутати користувачів при пошуку інформації та веб-майстрів при розміщенні контенту. Краще не використовувати цю структуру для створення великого сайту, оскільки на ньому розміщується багато гіперпосилань. Приклад структури сітки веб-сайту показаний на малюнку 2.3.

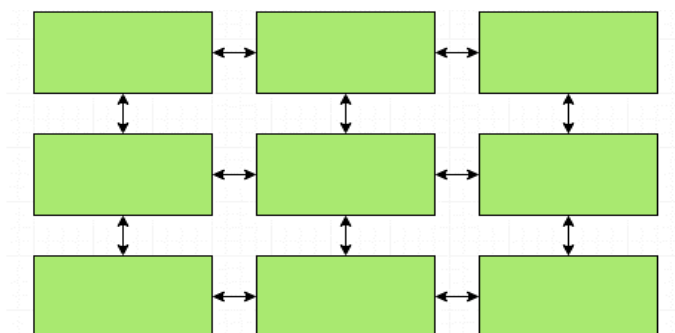


Рисунок 2.3 – Приклад схеми сітчастої структури сайту

Змішана структура - це складена структура, яка об'єднує 2 або більше структур зверху в 1. Всі сторінки цієї структури розміщуються з різним згладжуванням. Через складність реалізації ця структура рідко використовується при створенні сайту. Приклад схеми змішаної структури веб-сайту показаний на малюнку 2.4.

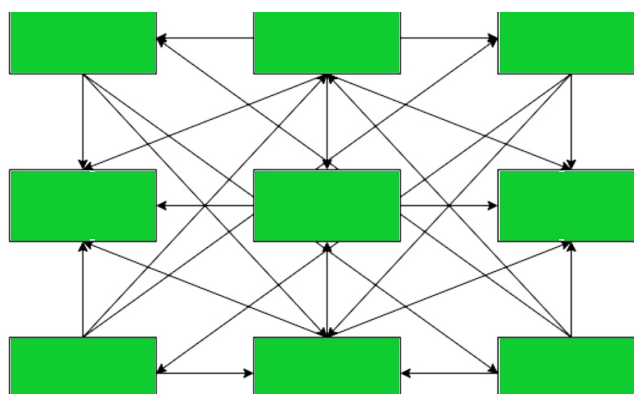


Рисунок 2.4 – Приклад схеми змішаної структури сайту

Після аналізу основних типів структур для створення веб-сайту була обрана деревоподібна структура, оскільки вона повністю відповідає вимогам

до створення веб-додатку для надання рекомендацій з вибору комп'ютерних комплектуючих. Структурна схема веб-додатку представлена на малюнку 2.5.

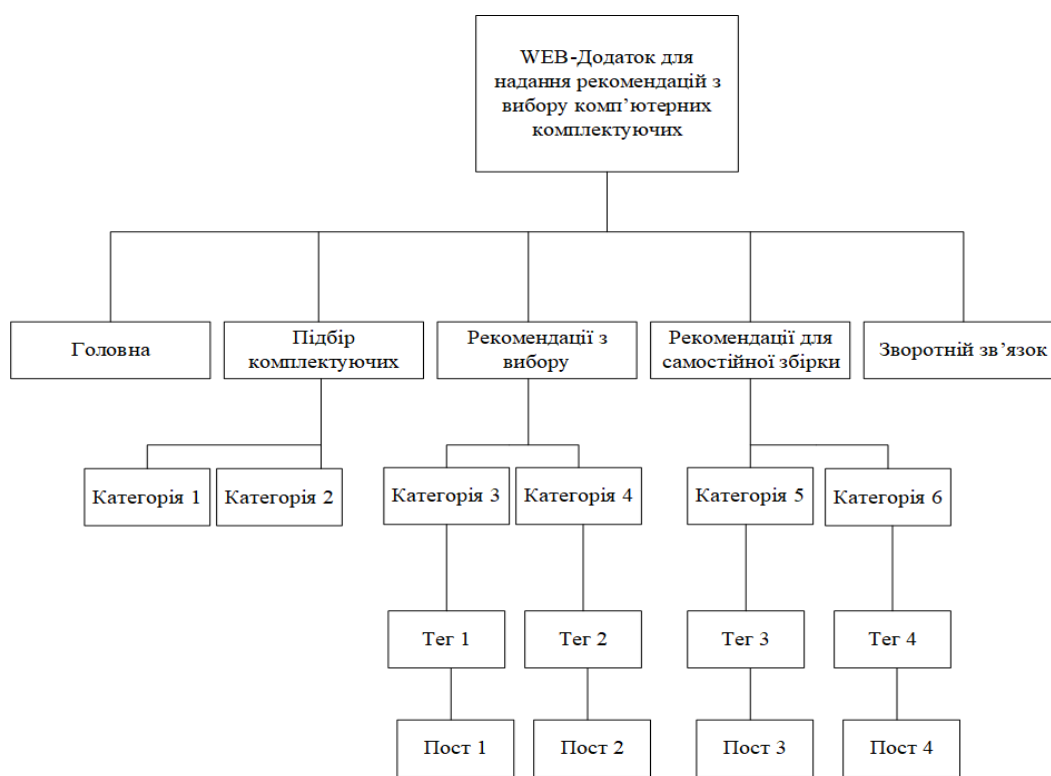


Рисунок 2.5 – Структурна схема веб-додатку

2.2 Розробка налаштувань та проектування панелей веб-додатку

Основною вимогою при створенні веб-програми, що надає рекомендації щодо вибору комп'ютерних компонентів, була можливість редагувати контент за допомогою адміністраторів без втручання програмістів. За допомогою CMS Wordpress можна зробити це прямо зараз для сторінок, які не мають спеціального дизайну, використовуючи редактор. Щоб відобразити спеціальний блок на сторінці, потрібно виконати деякі налаштування. Щоб ввести інформацію для нестандартного блоку, потрібно створити спеціальне поле у CMS Wordpress. Плагін Advanced custom field цілком підходить для цього. З його допомогою можна додавати додаткові поля для редагування сторінок у WordPress. ACF дозволяє менеджеру вмісту сайту змінювати вміст

готового блоку на панелі адміністрування сайту [11]. Які типи полів має ACF подано в таблиці 2.1.

Таблиця 2.1 - Типи полів Advanced custom field

Тип поля	Опис
Текст	Поле для введення тексту.
Текстова область	Основна текстова область для простих абзаців тексту.
Група кнопок	Інтерфейс для вибору значень.
Діапазон	Вибрати число між кінцевими точками.
Прапорець	Список для перевірки введених даних.
Перемикач	Список для вибору окремого значення.
Вибір	Вибирати один або кілька параметрів зі спадного списку.
True/False	Вибирати значення 1 або 0.
Файл	Завантажити та вибрати файл.
Галерея	Інтерфейс для управління колекцією вкладень.
Зображення	Завантажити та вибрати зображення через медіа-модаль WordPress.
oEmbed	Вставити відео, зображення, твіти, аудіо та багато іншого.
Редактор WYSIWYG	Редактор вмісту WordPress для редагування, стилізації тексту та медіа.
Вибір кольору	Вибирати шістнадцятковий колірний рядок.
Вибір дати	Спливаюче вікно вибору дати.
Вибір часу	Спливаюче вікно вибору часу.
Вибір дати та часу	Спливаюче вікно для вибору дати та часу.
Карти Google	Інтерфейс карти для вибору місць з автоматичним завершенням пошуку.
Акордеон	Об'єднання полів в складні панелі.
Клонувати	Відображає існуючі поля в базі даних без дублювання.
Гнучке вміст	Редактор на основі структурованих блоків.
Групувати	Структурувати поля в групи.
Повторювач	Повторює вміст (слайди, членів команди тощо).
Вкладка	Групує поля в розділи з вкладками.
Посилання	Вибирати або встановити посилання (URL, заголовок, ціль).
Посилання на сторінки	Спадне меню для вибору публікації, сторінки або URL-адреси архіву.
Таксономія	Вибір одного або декількох таксономічних термінів.

Групи полів можуть відображатися на певних сторінках веб-сайту, і доступ до них можливий з будь-якого місця коду веб-сайту. Вигляд головного вікна управління ACF в адмін-панелі зображено на рисунку 2.6.

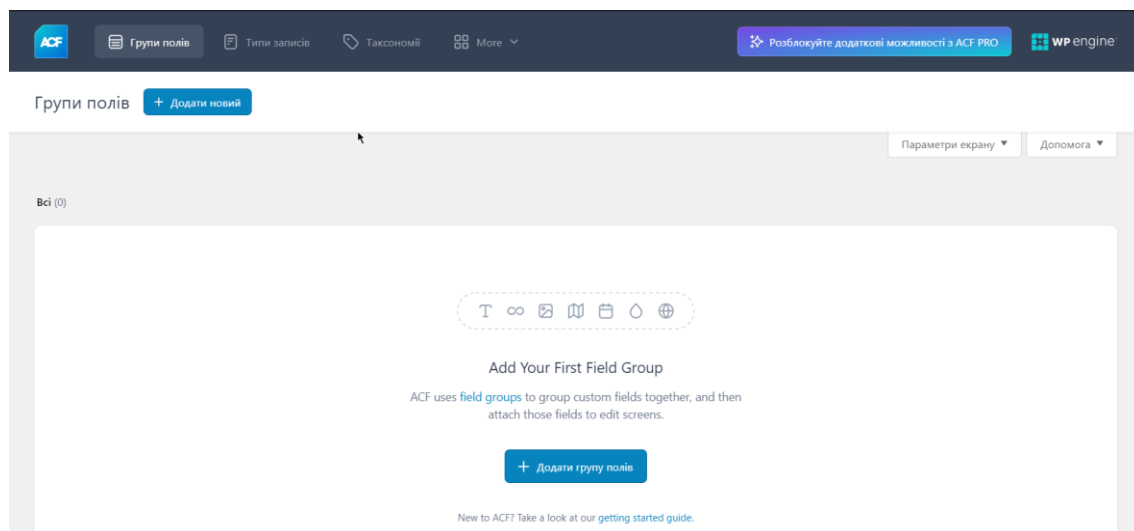


Рисунок 2.6 – Вигляд головного вікна управління ACF

Розробляючи веб-додаток, що надає рекомендації щодо вибору комп'ютерних компонентів, адміністратори можуть використовувати плагіни ACF, щоб забезпечити програмістам гнучкість та зручність управління вмістом, дозволяючи їм швидко оновлювати свої дані та надавати користувачам актуальну інформацію.

Таким чином, використання ACF у цьому проєкті значно покращило ефективність процесу управління вмістом, зробивши веб-додаток більш адаптивним та зручним для користувачів.

2.3 Розробка алгоритмів функціонування частин веб-додатку

Алгоритм функціонування являє собою чітко визначену послідовність кроків, що забезпечують безперебійну роботу та досягнення поставлених цілей системою. Він ґрунтується на базових алгоритмічних структурах: лінійній, розгалуженій та циклічній [12].

Лінійна структура передбачає послідовне виконання команд одна за одною, без пропусків чи повторень. Це найпростіша та зрозуміла структура, що використовується для виконання простих операцій.

Розгалужена структура передбачає залежність виконання команд від певної умови. Залежно від результату перевірки умови виконується одна з двох або більше послідовностей команд. Ця структура використовується для прийняття логічних рішень.

Циклічна структура передбачає багаторазове повторення певної послідовності команд до виконання заданої умови. Ця структура використовується для ітеративних операцій, що потребують багаторазового виконання одних і тих самих дій.

У контексті розробки форм та комунікацій на веб-сайті алгоритм функціонування описує логіку роботи цих елементів. Розглянемо алгоритм функціонування форми конфігуратора:

1. Користувач заповнює форму: Вводить свої дані та повідомлення.
2. Перевірка даних: Система перевіряє коректність введених даних (наприклад, формат запиту на підбір компонентів).
3. Відправка повідомлення:
 - a. Якщо дані коректні, формується запит та повідомлення надсилається на обробку.
 - b. Якщо дані некоректні, користувачеві відображається повідомлення про помилку.
 - c. Підтвердження: Користувачеві надсилається повідомлення про успішний (або невдалий) запит.

Переваги чітко описаного алгоритму:

- Зрозумілість: Алгоритм робить логіку роботи системи прозорою та зрозумілою для розробників, тестувальників та користувачів.

- Модифікованість: Чітко описаний алгоритм полегшує внесення змін та доопрацювання системи.
- Надійність: Алгоритм допомагає уникнути помилок та збоїв у роботі системи.
- Документація: Алгоритм може слугувати основою для технічної документації.
- Візуалізація алгоритму

Для кращого розуміння алгоритмів функціонування можна використовувати блок-схеми, діаграми потоків даних та інші методи візуалізації. Це допомагає наочно представити логіку роботи системи та полегшує її аналіз [13].

Блок-схема починається з початкового стану, позначеного як «Початок». Користувач заповнює форму, вводячи необхідні дані, цей етап представлений у вигляді прямокутника, що позначає дію користувача. Сформований запит надсилається на сервер для обробки, сервер обробляє отриманий запит, підбираючи відповідні компоненти потім повертає результати обробки запиту. В цей момент дані, введені користувачем, проходять перевірку на коректність, вузол перевірки зображений у вигляді ромба, який визначає логічну перевірку. Якщо дані коректні, алгоритм переходить до наступного етапу, а якщо дані некоректні, користувачеві відображається повідомлення про помилку і процес повертається до етапу заповнення форми. Це розгалуження також представлене ромбом. Після успішної перевірки даних користувач отримує повідомлення з результатами запиту. Алгоритм завершується, позначаючи кінець процесу, завершальний блок зображений як овал з написом "Кінець".

На рисунку 2.7 зображено блок-схему функціонування алгоритму форми.

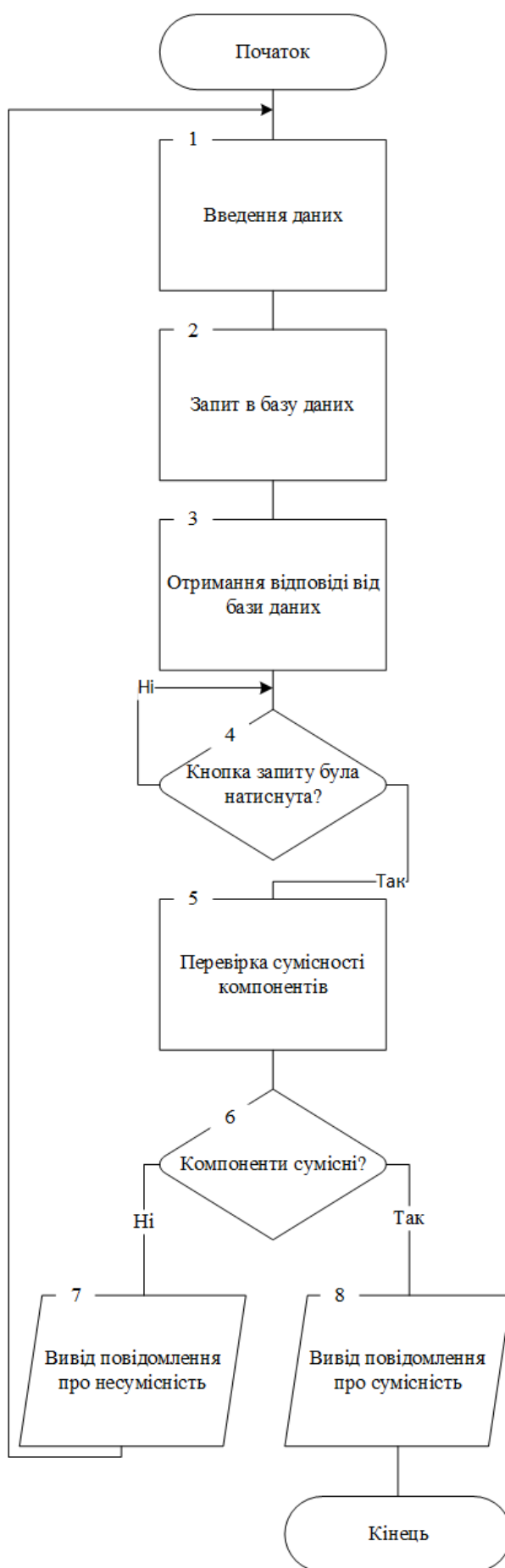


Рисунок 2.7 – Блок-схема алгоритму функціонування форми

Для кращого розуміння структури веб-додатку з надання рекомендацій для вибору комп'ютерних компонентів створено UML-діаграму класів, що зображена на рисунку 2.8.

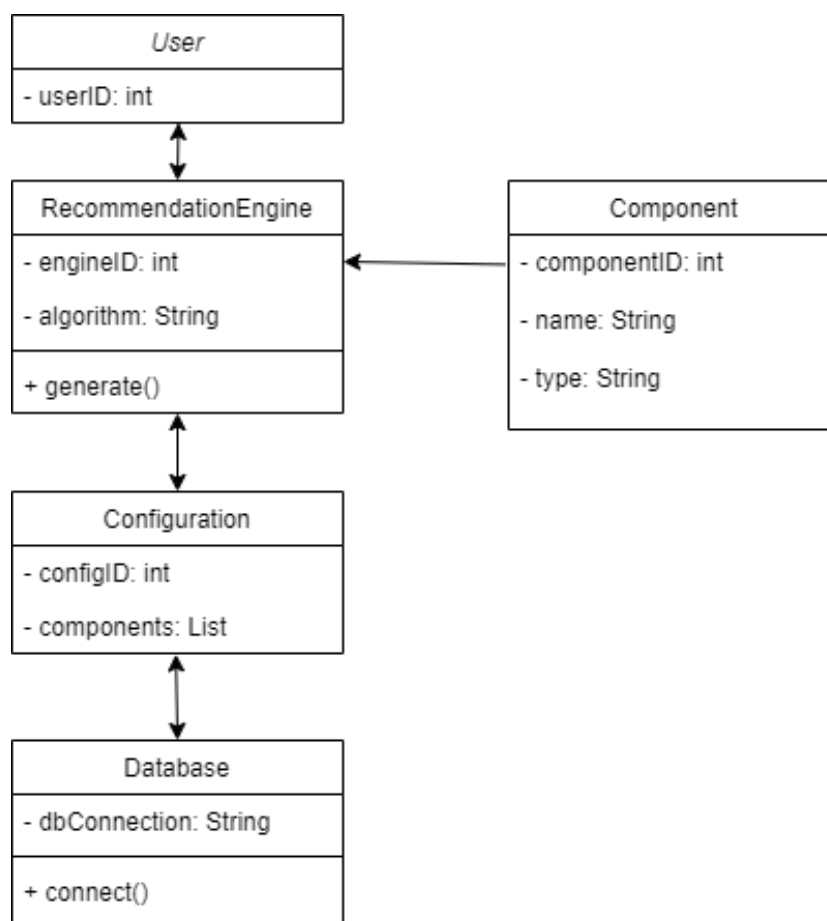


Рисунок 2.8 – UML-діаграма класів

На діаграмі класів зображені основні класи веб-додатку та їх взаємозв'язки. Вона включає класи «User», «RecommendationEngine», «Component», «Configuration» та «Database».

Класи:

- User: представляє користувача веб-додатку.
- RecommendationEngine: обробляє дані користувача та генерує рекомендації.
- Component: представляє окремий компонент комп'ютера.

- Configuration: зберігає конфігурації, створені користувачем.
- Database: управляє зберіганням даних.

ER-діаграма (ERD, Entity-Relationship Diagram) — це тип діаграми, що використовується для відображення логічної структури бази даних. Вона показує об'єкти даних (сутності) та відношення між ними. ER-діаграми допомагають проектувати та документувати бази даних, зокрема реляційні бази даних. ER-діаграма може бути корисною для відображення бази даних, яка зберігає інформацію про користувачів, комп'ютерні компоненти та конфігурації.

ER-діаграма відображає структуру бази даних нашого веб-додатку, яка включає сутності 'User', 'Configuration' та 'Component', а також взаємозв'язки між ними, що зображено на рисунку 2.9.

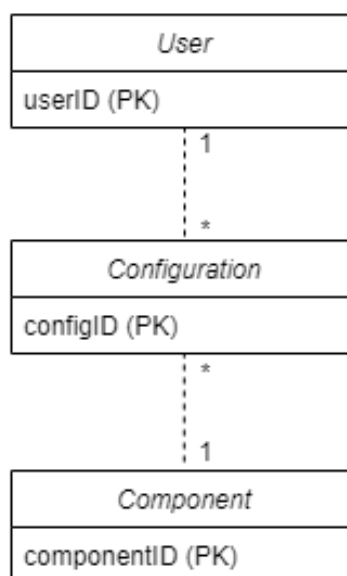


Рисунок 2.9 – ER-діаграма зв'язку між користувачем та базою даних

Валідація форм - це процес перевірки даних, введених користувачем у веб-формі, на предмет їх правильності та повноти [14]. Її мета - гарантувати, що дані відповідають очікуваному формату та правилам, перш ніж їх буде передано на сервер для обробки.

Переваги валідації форм:

- Зменшення помилок користувачів: Валідація допомагає користувачам уникнути помилок при заповненні форм, що економить їх час та зусилля.
- Покращення якості даних: Завдяки валідації дані, що надходять на сервер, мають кращу якість, що спрощує їх обробку та зберігання.
- Підвищення безпеки: Валідація може допомогти у виявленні та запобіганні шкідливим діям, наприклад, спробам введення недійсної інформації або втручання в роботу сайту.

Щоб забезпечити коректну роботу конфігуратора, необхідно реалізувати алгоритми перевірки сумісності компонентів. Дані про обрані компоненти надсилаються до бази даних для подальшої обробки. Вся інформація повинна зберігатися в адмін-панелі сайту, що дозволяє адміністратору контролювати та оновлювати дані відповідно до потреб користувачів.

2.4 Висновок розділу

Деревоподібна структура обрана з метою забезпечення чіткої ієрархії інформації, простоти навігації та зручності оновлення контенту. Ця структура відповідає принципам UI/UX дизайну та гарантує позитивний досвід користувачів.

Кастомні блоки на сторінках веб-додатку дають змогу динамічно генерувати контент, забезпечуючи гнучкість дизайну та простоту розробки. Цей підхід дозволяє створювати персоналізований досвід для кожного користувача та оптимізувати процес створення нових сторінок.

Реалізований алгоритм функціонування форми конфігуратора, що дає змогу користувачам не допустити помилок у сумісності компонентів та отримувати вказівки з їх вибору. Крім того, алгоритм автоматично перевіряє кожен вибраний компонент на сумісність з іншими, забезпечуючи цілісність і

коректність зібраної системи. У випадку виявлення несумісності, система надає детальні повідомлення про помилку та пропонує альтернативні варіанти, що значно спрощує процес вибору комплектуючих навіть для користувачів з обмеженим технічним досвідом.

Переваги:

- Зручний інтерфейс: Деревоподібна структура та кастомні блоки забезпечують інтуїтивно зрозумілу навігацію та приємний візуальний дизайн.
- Гнучкість: Динамічний контент та налаштовувані блоки дозволяють легко оновлювати інформацію та створювати персоналізований досвід для користувачів.
- Функціональність: Контактна форма, система коментарів та модерація контенту гарантують ефективну комунікацію з користувачами та безпеку веб-додатку.

Загалом, веб-додаток вирізняється чітко продуманою структурою, динамічним контентом та зручними функціональними можливостями, що робить його цінним інструментом для підбору комп'ютерних комплектуючих.

3 РОЗРОБКА ЧАСТИН WEB-ДОДАТКУ

3.1 Вибір інструментів розробки веб-додатку

Цей розділ описує технології, які використовуються для розробки веб-сайту на основі CMS WordPress.

Ядро WordPress написано на PHP, тому ця мова є основою для створення динамічних сторінок, шаблонів тем та плагінів [15]. PHP використовується на стороні сервера для генерування контенту та обробки даних користувачів. JavaScript: JavaScript використовується для додавання інтерактивності та динамічної поведінки на веб-сторінки. За допомогою JavaScript можна створювати слайдери, карти, форми зворотного зв'язку, анімації та інші функціональні елементи. HTML використовується для створення базової структури та основного контенту веб-сторінок. Ця мова описує, як текст, зображення та інші елементи повинні відображатися в браузері користувача. CSS використовується для стилізації зовнішнього вигляду веб-сторінок. За допомогою CSS можна контролювати шрифти, кольори, розміри елементів, розширення та інші візуальні аспекти веб-сайту.

WordPress використовує базу даних MySQL для зберігання контенту, налаштувань та інших даних веб-сайту. Веб-сервер - це програмне забезпечення, яке зберігає файли веб-сайту та робить їх доступними для користувачів через Інтернет. Apache та Nginx - два популярних веб-сервери, які використовуються для хостингу веб-сайтів на WordPress.

Технології, що використовуються для розробки WordPress, є широко поширеними та добре підтримуються. Ці технології надають розробникам гнучкість для створення веб-сайтів з різними функціональними можливостями та дизайном. Завдяки великій спільноті користувачів та розробників WordPress, доступно багато ресурсів та підтримки для цих технологій.

Не обов'язково володіти глибокими знаннями всіх перелічених технологій для роботи з WordPress. Початківці можуть використовувати

візуальний інтерфейс адміністрування WordPress для створення базових веб-сайтів без коду. Досвідчені розробники можуть використовувати розширені можливості WordPress для створення складних та функціональних веб-сайтів з унікальним дизайном. Використання PHP, JavaScript, HTML та CSS у розробці веб-сайту на WordPress дає змогу створювати сучасні, динамічні, візуально привабливі та зручні веб-ресурси [16].

WordPress використовує PHP, мову програмування для роботи з веб-серверами, для динамічного створення веб-сторінок з бази даних. Це робить його більш потужним, ніж статичний веб-сайт, де кожна сторінка створюється вручну.

WordPress PHP виконує кілька важливих функцій. Це дозволяє отримувати з бази даних таку інформацію, як записи повідомлень, сторінки та імена користувачів. Дані з бази даних використовуються для створення динамічного HTML-коду, який формує веб-сторінку, що відображається Користувачем. PHP також використовується для створення шаблонів і плагінів, які розширюють функціональність WordPress, дозволяючи створювати унікальні дизайни та додавати нові функції, не змінюючи базовий код WordPress. Крім того, PHP обробляє запити користувачів, такі як відкриття сторінки, відправка форми, завантаження файлу і т.д., і перевіряє права доступу користувача.

Використання PHP у WordPress має багато переваг. PHP робить WordPress дуже гнучким і дозволяє створювати різні веб-сайти. Завдяки PHP веб-сайти WordPress можуть динамічно генерувати вміст, що робить їх більш інтерактивними та зручними для користувачів. Численні плагіни та теми WordPress, розроблені за допомогою PHP, роблять цю платформу дуже розширюваною. Крім того, існує багато ресурсів для вивчення PHP, що дозволяє розробникам усіх рівнів отримати доступ до PHP.

JavaScript також відіграє важливу роль у WordPress, додаючи інтерактивності на стороні клієнта. Наприклад, ви можете використовувати JavaScript для створення динамічного меню, яке розкривається або ховається

при наведенні курсору. Ви можете перевірити дані, введені користувачем у форму, створити анімацію та інші візуальні ефекти, а також додати новий вміст на сторінку, не перезавантажуючи всю сторінку.

Таким чином, PHP та JavaScript відіграють важливу роль у роботі WordPress. PHP використовується для динамічного створення веб-сторінок з бази даних, а JavaScript – для додавання інтерактивності на стороні клієнта. Це поєднання робить WordPress потужною та гнучкою платформою для створення веб-сайтів. [17].

Переваги мови JavaScript:

- Швидкість: JavaScript забезпечує швидку компіляцію завдяки своїй імперативній природі. Оскільки це мова сценаріїв на стороні клієнта, яка працює в браузері користувача, це економить час при завантаженні сторінок із сервера та пришвидшує роботу Програм у JavaScript.
- Простий у використанні: JavaScript інтерпретується веб-браузером, тому не потрібно встановлювати компілятор. Це полегшує запуск коду JavaScript без будь-яких додаткових налаштувань. Крім того, відносно простий синтаксис мови полегшує його вивчення новачками.
- Багатосторонність: JavaScript підтримується всіма сучасними веб-браузерами і може використовуватися на різних платформах, включаючи ПК, мобільні пристрої та планшети.
- Спільнота та підтримка: Завдяки популярності JavaScript він має велику та активну спільноту розробників. Це дозволяє легко відповідати на запитання та допомагати у вирішенні проблем.
- Інтеграція: JavaScript добре інтегрується з іншими мовами програмування, що робить його корисним для розробки веб-сайтів та веб-додатків. Його можна вбудувати в будь-яку веб-сторінку, що робить його гнучким інструментом для додавання інтерактивного та динамічного вмісту.

- Інтерактивність: JavaScript надає широкий спектр інтерфейсів для створення динамічних та інтерактивних веб-сторінок. JavaScript можна використовувати для створення більш привабливого та зручного у використанні веб-сайту.

JavaScript – це потужна та універсальна мова програмування, яка пропонує багато переваг розробникам веб-сайтів та веб-додатків. Його швидкість, простота використання, універсальність, велика спільнота та широкі можливості інтеграції роблять його цінним інструментом для створення сучасних динамічних веб-сторінок.

CSS (Cascading Style Sheets) – це мова для опису зовнішнього вигляду веб-сторінок, створених з використанням HTML. CSS є однією з 3 основних технологій веб-розробки, поряд з HTML і JavaScript.

Функції CSS включають відокремлення вмісту від дизайну, що робить код більш зрозумілим, гнучким та керованим. CSS дозволяє керувати зовнішнім виглядом елементів HTML, таких як колір, шрифт, розмір, розташування, межі та вставка зображень. Ви також можете створювати адаптивні веб-сайти, які автоматично підлаштовуються під різні розміри екрану, що робить їх зручними для перегляду на комп'ютерах, планшетах і мобільних пристроях. Крім того, CSS економить час і код, оскільки стилі можна визначити один раз і застосувати до кількох елементів на різних веб-сторінках. Ви також можете підвищити продуктивність свого веб-сайту, зменшивши обсяг завантажуваного HTML-коду.

HTML (мова гіпертекстової розмітки) - це мова текстової розмітки, яка використовується для створення структури та вмісту веб-сторінок. HTML описує вміст веб-сторінки, але не визначає, як вона виглядає.

Основні елементи HTML містять теги, які є будівельними блоками html і використовуються для вказівки різних елементів веб-сторінки, таких як заголовки, абзаци, зображення та посилання. Атрибути надають додаткову інформацію про теги. Наприклад, тег «img» використовується для вставки зображення, а атрибут «src» - для визначення URL-адреси зображення.

Семантичні теги описують зміст і призначення контенту і допомагають пошуковим системам краще розуміти зміст веб-сторінок і ранжувати їх в результатах пошуку.

CSS та HTML - це дві додаткові мови, які є основою для створення веб-сторінок. CSS використовується для управління зовнішнім виглядом веб-сторінки, а HTML - для створення її структури та вмісту. Разом вони дозволяють розробникам створювати красиві, зручні для користувача та доступні веб-сайти.

Середовищами розробки були обрані Brackets для фронтенду та PhpStorm для бекенду. Brackets - це безкоштовний та відкритий текстовий редактор від Adobe, спеціально розроблений для front-end розробки. Він пропонує широкий спектр інноваційних функцій та можливостей, що робить його цінним інструментом для створення сучасних веб-сайтів та веб-додатків. Вікно редактору Brackets показано на рисунку 3.1.

Основні переваги Brackets:

- Live Edit: Ця унікальна функція дозволяє бачити зміни, внесені до HTML, CSS та JavaScript, безпосередньо у вашому браузері, без необхідності перезавантажувати сторінку. Це значно економить час та зусилля під час розробки та тестування.
- Live Preview: Переглядайте зміни в дизайні вашого сайту в реальному часі, просто навівши курсор на код. Це дає вам чітке уявлення про те, як ваші зміни впливатимуть на візуальний аспект сайту, без необхідності постійно перемикатися між редактором та браузером.
- Editor Preview: Ця функція розділяє ваш екран на дві частини, показуючи код з одного боку та візуальне представлення вашого сайту з іншого. Це полегшує одночасну роботу з кодом та його візуальним ефектом, роблячи процес розробки більш ефективним.
- Підтримка препроцесорів: Brackets підтримує популярні препроцесори CSS, такі як LESS та SCSS, що дозволяє вам писати

код у більш зрозумілому та структурованому форматі, роблячи його більш читабельним та зручним для обслуговування.

- Розширюваність: Завдяки величезній бібліотеці плагінів, доступних в магазині Extension Registry, ви можете значно розширити можливості Brackets, додаючи нові функції, інтегруючи інші інструменти та налаштовуючи інтерфейс редактора відповідно до ваших потреб.
- Підтримка різних платформ: Brackets доступний для Windows, macOS та Linux, що робить його зручним інструментом для розробників, які працюють на різних операційних системах.

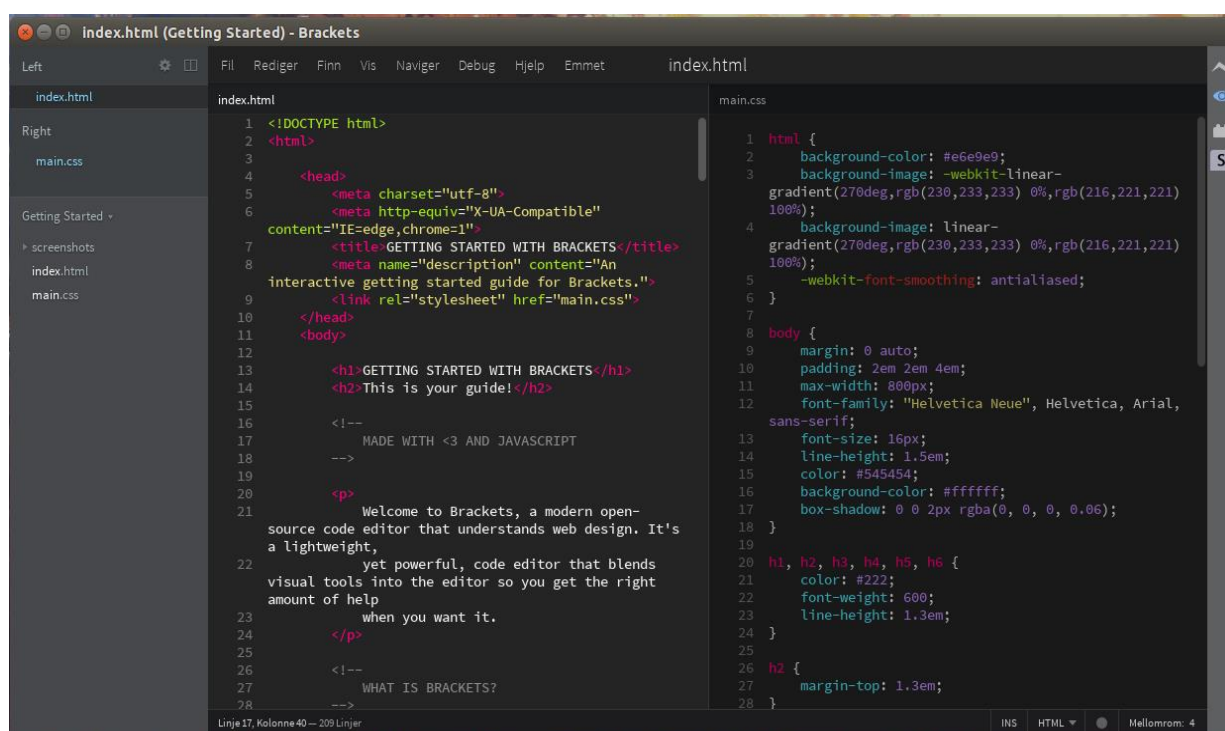


Рисунок 3.1 – Приклад вікна редактору Brackets

Brackets – це чудовий вибір для веб-розробників будь-якого рівня досвіду, як для початківців, так і для досвідчених фахівців. Його інноваційні функції, простота використання та гнучкість роблять його цінним інструментом для створення та редагування веб-сайтів та веб-додатків. [18].

PhpStorm – це вдосконалене інтегроване середовище розробки (IDE) від JETBRAINS, спеціально розроблене для PHP [19]. Створений на базі платформи IntelliJ IDEA, цей потужний інструмент пропонує широкий спектр функцій, які допоможуть PHP-розробникам усіх рівнів значно підвищити продуктивність і якість коду. PhpStorm володіє глибоким розумінням коду на PHP, HTML і JavaScript, забезпечуючи оперативний аналіз, запобігання помилок і автоматичний рефакторинг. Це економить час і зусилля розробників і дозволяє їм писати код швидше та ефективніше. PhpStorm підтримує такі функції, як генератори, спільні програми, простори імен, закриття, типи та синтаксис коротких масивів, що охоплюють як сучасні, так і традиційні проекти PHP. Завдяки вбудованому редактору SQL ви можете редагувати результати запитів до бази даних безпосередньо в IDE. Завдяки платформі IntelliJ IDEA, PhpStorm можна розширювати за допомогою плагінів від JetBrains і плагінів, розроблених користувачами. Це дозволяє налаштувати IDE відповідно до ваших потреб та уподобань. Переваги використання PhpStorm:

- Покращена продуктивність: розумні функції PhpStorm допомагають писати код швидше та з меншою кількістю помилок, заощаджуючи час та зусилля розробників.
- Покращена якість коду: PhpStorm постійно аналізує код, виявляє потенційні проблеми та пропонує кращі рішення, щоб зробити код чистішим, надійнішим та керованим.
- Ефективна розробка: широкий спектр інструментів PhpStorm, включаючи налагодження, тестування та профілювання, полегшує розробку та налагодження складних PHP-додатків.
- Підтримка популярних фреймворків: PhpStorm підтримує популярні фреймворки PHP, такі як Laravel, Symfony та Yii, забезпечуючи зручне середовище для розробки.

На рисунку 3.2 можна бачити головне вікно програми PhpStorm

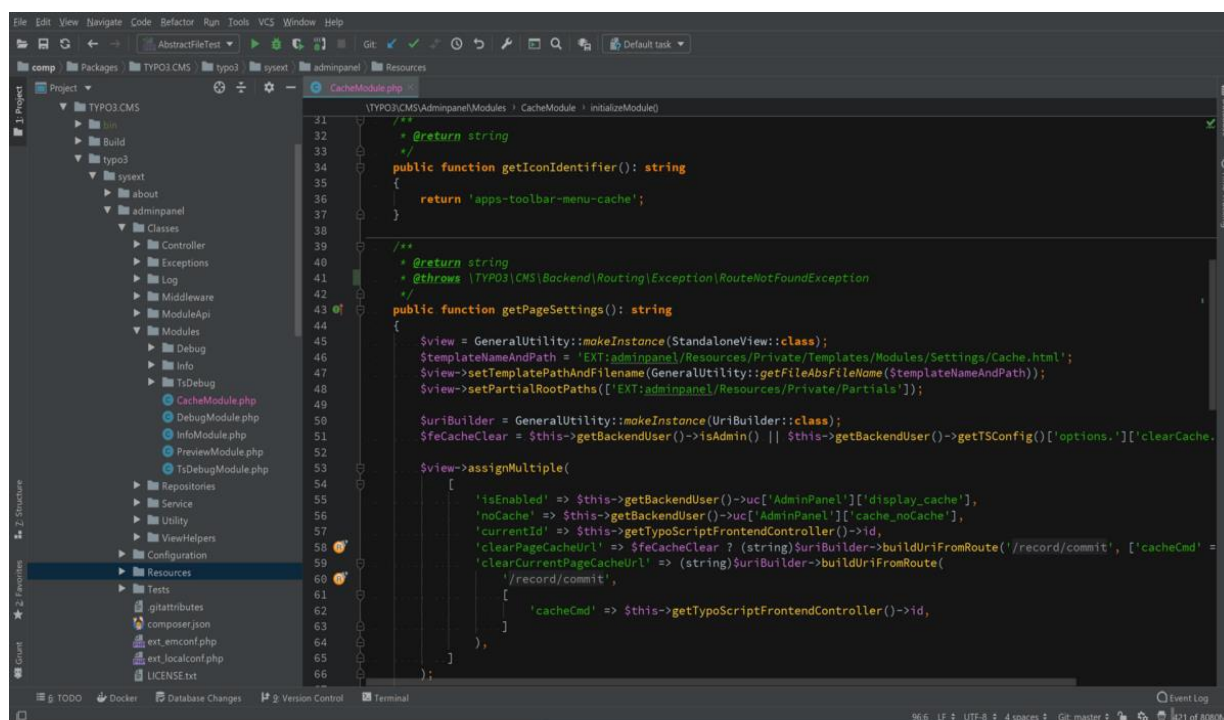


Рисунок 3.2 – Вікно програми PhpStorm

PhpStorm – хороший вибір для розробників PHP, які прагнуть підвищити продуктивність, команди, які хочуть покращити якість коду, компанії, які потребують ефективних інструментів розробки та налагодження, розробники, які використовують найновіший фреймворк PHP.

3.2 Реалізація форми конфігуратора

Для розробки форми конфігуратора у CMS Wordpress необхідно використати всі наявні засоби, адже у даній роботі це найголовніший модуль функціональності веб-додатку. Створення форми конфігуратора в CMS WordPress може здатися складним завданням, але за допомогою доступних інструментів та плагінів його можна виконати ефективно. Для цього необхідно опиратися на попередній розділ, у якому було розглянуто алгоритм роботи програми та необхідно визначити функціональні можливості, метод створення форми, створити та налаштувати та стилізувати форму, яку потім необхідно буде з'єднати з бекендом. Поля для взаємодії з користувачем мають містити

всі типи компонентів, щоб користувач міг обирати їх із списку комплектуючих, які будуть міститися у базі даних. Для перевірки валідації правильно підібраних компонентів буде використовуватися перевірка на сумісність за характеристиками та технологіями. Вибрані дані мають порівнюватися за допомогою блоків коду на PHP для динамічності відображення та взаємодією з базою даних MySQL [20].

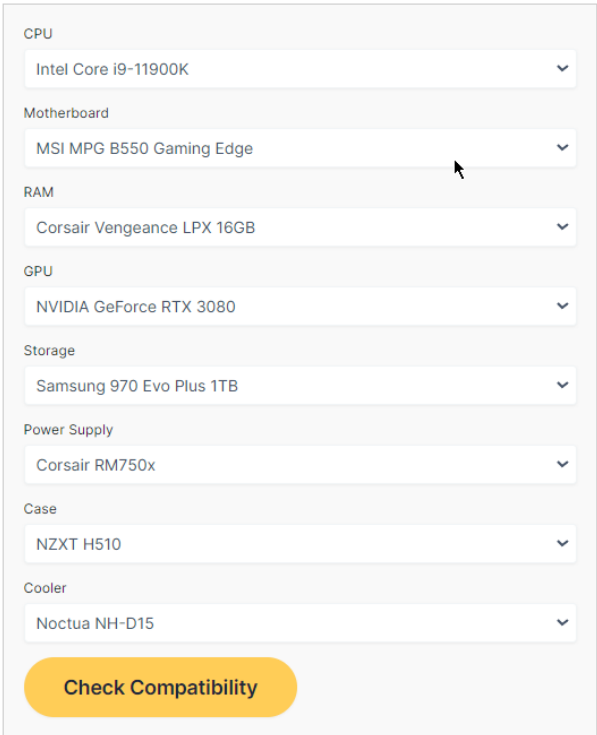
```
// Перевірка сумісності CPU і материнської плати

if ($cpu && $motherboard) {
    global $wpdb;
    $cpu_data = $wpdb->get_var($wpdb->prepare(«SELECT compatibility_data
FROM wp_pc_parts WHERE id = %d AND type = 'cpu'«, $cpu));
    $motherboard_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'motherboard'«,
$motherboard));
    if ($cpu_data && $motherboard_data) {
        $cpu_data = json_decode($cpu_data, true);
        $motherboard_data = json_decode($motherboard_data, true);
        if (!in_array($cpu_data['socket'], explode(',', $motherboard_data['socket'])))
        {
            $compatible = false;
            $messages[] = «CPU and Motherboard are not compatible.»;
        }
    }
}
```

Як можна спостерігати з цього блоку коду, де виконується перевірка сумісності процесора та материнської плати було використано PHP, яка бере інформацію для порівняння з бази даних в стовпці compatibility. Аналогічний метод перевірки необхідно також реалізувати і для всіх інших компонентів, враховуючи всі наявні технічні характеристики, для сумісності не тільки між материнською платою, а і для сумісності з усіма іншими компонентами збірки. Використовуючи обраний метод, створимо

структуру форми, додаючи необхідні поля, такі як текстові поля, випадаючі списки та кнопки. Вигляд форми можна побачити на рисунку 3.2.

Підібрати комплектуючі



CPU
Intel Core i9-11900K

Motherboard
MSI MPG B550 Gaming Edge

RAM
Corsair Vengeance LPX 16GB

GPU
NVIDIA GeForce RTX 3080

Storage
Samsung 970 Evo Plus 1TB

Power Supply
Corsair RM750x

Case
NZXT H510

Cooler
Noctua NH-D15

Check Compatibility

CPU and Motherboard are not compatible.

Рисунок 3.2 – Вигляд форми конфігуратора

Як можна спостерігати з форми, внизу має виводитися повідомлення про сумісність компонентів, є добре видна кнопка для перевірки, форма знаходиться вище для зручності, всі типи компонентів розташовані в порядку їх важливості в збірці, Безумовно всі компоненти є досить важливими, але без кулера зібраний комп'ютер має ввімкнутися, хоч і не на тривалий час.

Використовуючи CSS було налаштовано зовнішній вигляд форми, роблячи її візуально привабливою та зручною для користувачів, форма добре виглядає та правильно працює на різних пристроях, включаючи мобільні телефони та планшети.

Залежно від обраного методу створення форми, а саме форма написана на PHP з використанням JavaScript, HTML та CSS, необхідно зв'язати її з

backend-логікою веб-додатку. Це може включати збереження даних форми в базі даних, надсилання електронних листів або виконання інших дій. Також до цієї форми необхідно в обов'язковому порядку підключити бази даних MySQL для правильної роботи. В базі даних буде занесена інформація про всі наявні комплектуючі.

3.3 Реалізація розділів веб-додатку

Важливим завданням також є наповнення сайту, створення додаткових розділів та його вигляд. Так як сайт має основне завдання в наданні рекомендацій з вибору комп'ютерних комплектуючих то основним на сайті має бути додаток з надання рекомендацій, а саме конфігуратор, тому посилання, у вигляді кнопки, на нього необхідно розмістити на головній сторінці, сама головна сторінка має градієнтний фон, тематичне зображення та основну інформацію, рухатися по сайті можна як опустивши вниз, так і за посиланням у навігаційному меню. На рисунку 3.3 зображено головний вигляд головної сторінки.

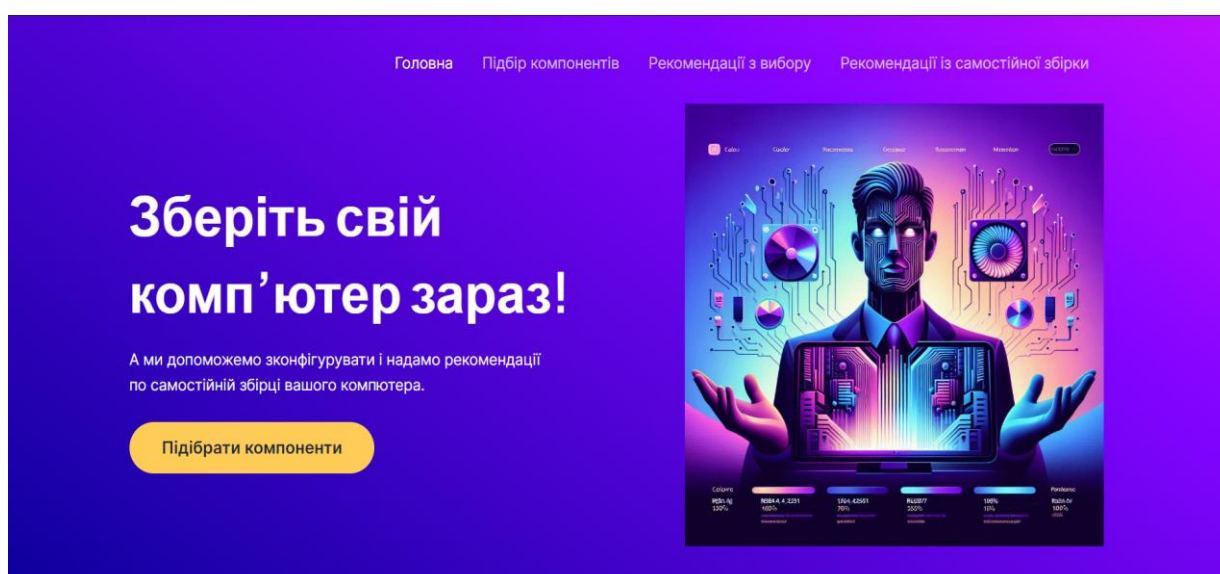


Рисунок 3.3 – Головна сторінка WEB-додатку

Наступною сторінкою є «Рекомендації з самостійної збірки». На цій сторінці детально описано принцип та послідовність дій при самостійній збірці, також описані типові помилки при самостійній збірці комп'ютерів. На рисунках 3.4 та 3.5 зображено вигляд цих сторінок.

Потенційні помилки при самостійному підборі компонентів ПК

Збирання власного персонального комп'ютера (ПК) може бути привабливим варіантом для користувачів, які прагнуть оптимізувати конфігурацію під конкретні потреби та отримати унікальну систему. Однак, під час вибору компонентів існує ряд потенційних помилок, що можуть призвести до нестабільної роботи, зниження продуктивності або навіть пошкодження обладнання. Розглянемо ключові аспекти, на які слід звернути увагу, щоб уникнути подібних негативних наслідків. Несумісність компонентів є, мабуть, найкритичнішою помилкою. Забезпечення сумісності є фундаментальним кроком у створенні функціональної системи. Необхідно ретельно перевірити сумісність материнської плати з обраним процесором, оперативну пам'ять підібрати відповідно до типу роз'ємів на материнській платі, а блок живлення повинен мати достатню потужність для живлення усієї системи. Для полегшення цього завдання рекомендується	Нехтування балансом компонентів може призвести до неефективності системи. Наприклад, немає сенсу інвестувати в потужний ігровий процесор разом із слабкою відеокартою або систему охолодження, яка відповідає навпаки. Значно ефективніше зосередитися на збалансованій конфігурації, яка відповідає основним завданням, для яких призначається ПК. Недооцінка потреб користувача є поширеною помилкою, що призводить до неефективного використання ресурсів. Перед вибором компонентів необхідно чітко визначити цільове призначення комп'ютера. Потужний ігровий ПК буде зайвим для офісної роботи з документами, а слабкий варіант не впорається з сучасними іграми. Перевитрати бюджету можуть стати наслідком захоплення ідеєю самостійної збірки. Необхідно враховувати усі витрати, що включають не тільки вартість компонентів, але й периферійні пристрої, операційну систему та програмне забезпечення. Іноді придбання готового ПК певної конфігурації	Недостатня увага до системи охолодження може призвести до перегріву компонентів та їх потужності процесора. Для високопродуктивних компонентів може знадобитися додаткове охолодження, таке як рідинне. Недооцінювання ролі блоку живлення є серйозною помилкою. Блок живлення є критично важливим компонентом, що забезпечує живлення усієї системи. Обирайте блок живлення з достатнім запасом потужності та від відомого виробника з гарною репутацією. Неякісний блок живлення може не тільки не забезпечити стабільну роботу системи, але й потенційно пошкодити інші компоненти. Нехтування гарантійними зобов'язаннями може призвести до додаткових витрат у разі несправності компонентів. Переконайтеся, що на всі придбані компоненти поширюється гарантія виробника. Ігнорування відгуків користувачів може позбавити вас цінної інформації. Перед
---	---	---

Рисунок 3.4 – Вигляд сторінки з потенційними проблемами при самостійній збірці

На цьому рисунку добре видно описані потенційні проблеми, які виникають при самостійній збірці. Під час складання комп'ютера користувач може зіткнутися з низкою складнощів. потенційні проблеми підкреслюють важливість уважного планування та ретельної перевірки всіх компонентів перед їх установкою. Використання рекомендаційної системи може значно

спростити процес вибору сумісних компонентів та мінімізувати ризики виникнення таких проблем.

Рекомендації із самостійної збірки ПК

З чого почати збірку комп'ютера:

- **Визначте бюджет та потреби:** Скільки ви готові витратити? Які завдання буде виконувати комп'ютер?
- **Підберіть компоненти:** Процесор, материнська плата, оперативна пам'ять, відеокарта, жорсткий диск/SSD, блок живлення, корпус.
- **Зберіть комп'ютер:** Дотримуйтеся інструкцій та будьте обережні.
- **Встановіть операційну систему та драйвери.**

Детальніше:

1. Планування:

- **Бюджет:** Визначте, скільки ви можете витратити на компоненти.

3. Збірка:

- **Дотримуйтеся інструкцій:** Кожен компонент має інструкцію з його встановлення.
- **Будьте обережні:** Не торкайтеся до контактів компонентів та не використовуйте силу.
- **Заземлення:** Використовуйте антистатичний браслет, щоб уникнути пошкодження компонентів статичною електрикою.

4. Встановлення програмного забезпечення:

- **Операційна система:** Windows, Linux, macOS.
- **Драйвери:** Необхідні для роботи компонентів.



Рисунок 3.5 – Вигляд сторінки із рекомендаціями із самостійної збірки

У навігаційному меню є кнопка «Рекомендації з вибору», яка веде на наступну сторінку, де описані рекомендації з вибору типу комп'ютера. В цьому меню описано певні основні типи комп'ютерів, які використовуються, деякі їх характеристики та приблизний вигляд на фотографіях. Також присутня кнопка «Вгору» в правій нижній частині сайту для зручності переміщення сайтом. На рисунку 3.6 зображено вигляд рекомендацій з вибору.

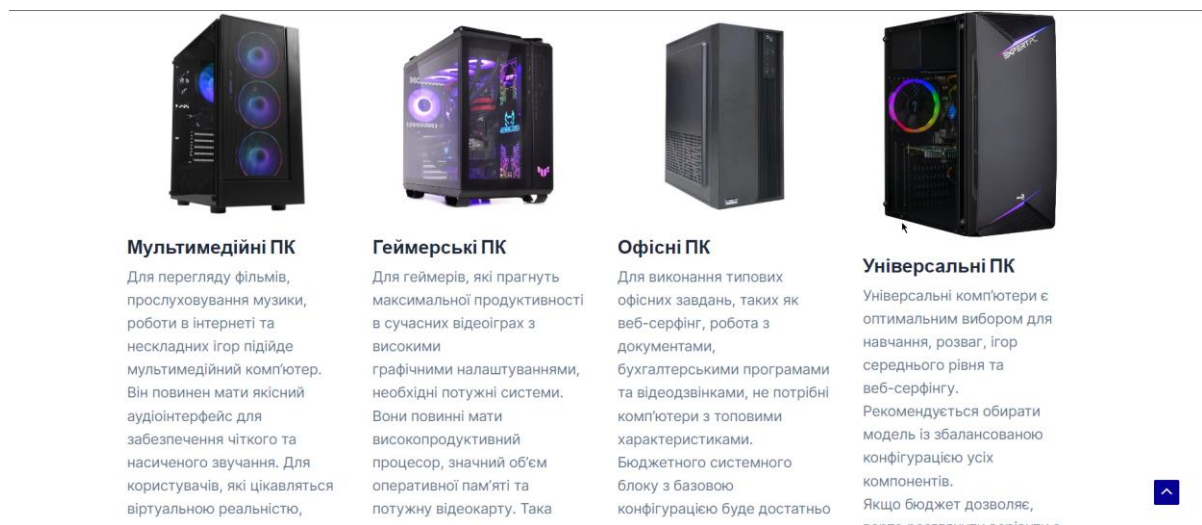


Рисунок 3.5 – Вигляд сторінки надання рекомендацій з вибору

На цьому рисунку добре видно описані потенційні проблеми, які виникають при самостійній збірці.

Рекомендації із самостійної збірки ПК

З чого почати збірку комп'ютера:

- **Визначте бюджет та потреби:** Скільки ви готові витратити? Які завдання буде виконувати комп'ютер?
- **Підберіть компоненти:** Процесор, материнська плата, оперативна пам'ять, відеокарта, жорсткий диск/SSD, блок живлення, корпус.
- **Зберіть комп'ютер:** Дотримуйтеся інструкцій та будьте обережні.
- **Встановіть операційну систему та драйвери.**

Детальніше:

1. Планування:

- **Бюджет:** Визначте, скільки ви можете витратити на компоненти.

3. Збірка:

- **Дотримуйтеся інструкцій:** Кожен компонент має інструкцію з його встановлення.
- **Будьте обережні:** Не торкайтеся до контактів компонентів та не використовуйте силу.
- **Заземлення:** Використовуйте антистатичний браслет, щоб уникнути пошкодження компонентів статичною електрикою.

4. Встановлення програмного забезпечення:

- **Операційна система:** Windows, Linux, macOS.
- **Драйвери:** Необхідні для роботи компонентів.



Рисунок 3.6 – Вигляд сторінки із рекомендаціями із самостійної збірки

3.4 Тестування веб-додатку

Тестування програмного забезпечення (Software Testing) – це метод контролю якості, що включає планування тестових дій, розробку тестів, їх виконання та аналіз отриманих результатів [21]. Тестування може бути ручним і автоматизованим.

Ручне тестування включає виконання тестів тестувальниками без застосування автоматизованих інструментів. Це найбільш простий і базовий тип тестування, оскільки не потребує значних додаткових знань [22]. Переваги ручного тестування:

- Тестувальник може виявити нові проблеми, слідуючи абстрактним крокам сценарію тесту.
- Це дешевший вид тестування.

Автоматизоване тестування використовує спеціальне програмне забезпечення для виконання тестів і порівняння очікуваних та фактичних результатів. Воно автоматизує повторювані дії, що необхідні для досягнення максимального тестового покриття. Переваги автоматизованого тестування:

- Висока швидкість.
- Можливість повторного використання тестів.
- Можливість написання різних тест-кейсів.
- Працює в режимі 24/7 без нагляду, надаючи детальні результати.
- Висока точність без людських помилок.
- Виключає необхідність виконання монотонних операцій.

Автоматизовані тести допомагають оптимізувати роботу тестового відділу, скорочуючи час на проведення тестів, оскільки один тест можна використовувати для кількох проектів. Ручне тестування найбільше використовується для перевірки нових функцій і в дослідницькому тестуванні для виявлення проблем у складних сценаріях користувача.

Автоматизоване тестування не замінює ручне, а доповнює його. У багатьох аспектах взаємодії з сайтом людські очі та руки залишаються незамінними, але автоматизовані тести значно економлять час. Вибір типу тестування залежить від конкретних задач, причому їх можна комбінувати у пропорціях, що найкраще підходять для проекту.

Для даного веб-додатку було обрано саме ручне тестування, оскільки цей підхід дозволяє тестувальнику безпосередньо взаємодіяти з додатком та інтуїтивно виявляти можливі проблеми. Процес ручного тестування включає кілька етапів. На першому етапі здійснюється планування тестування, яке передбачає визначення цілей тестування та складання тест-плану. У тест-плані зазначаються перелік функціональних вимог та сценаріїв тестування. Далі розробляються тест-кейси для кожної функції веб-додатку. Тест-кейс – це опис дій, які необхідно виконати для перевірки коректності роботи певної функції, а також визначення очікуваних результатів для кожного тест-кейсу. Після розробки тест-кейсів тестувальник приступає до виконання тестів. Це включає ручне виконання дій, описаних у тест-кейсах, та фіксування фактичних результатів. Фактичні результати порівнюються з очікуваними для виявлення можливих розбіжностей. Завершальним етапом є аналіз та звітування.

Тестувальники збирають дані про всі знайдені дефекти, створюють звіт про тестування, що включає опис знайдених проблем, їх вплив на функціональність та пропозиції щодо їх усунення.

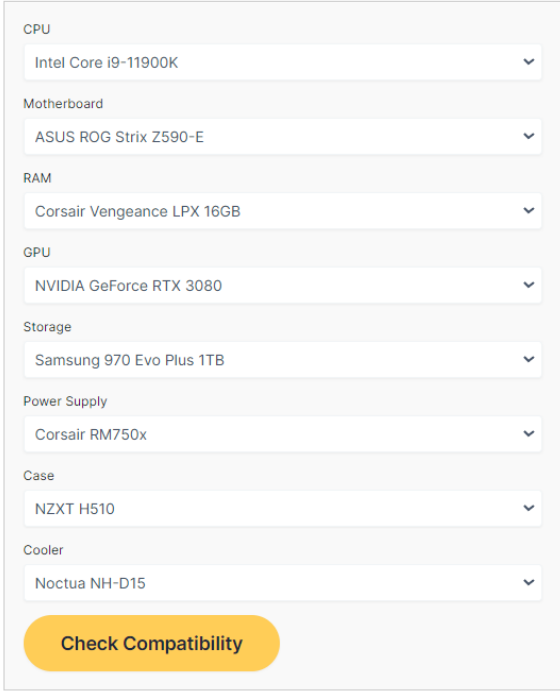
Ручне тестування має кілька переваг. Воно забезпечує гнучкість, дозволяючи тестувальникам адаптувати тестові сценарії на ходу, в залежності від реакції системи. Тестувальники також можуть використовувати свій досвід і інтуїцію для виявлення проблем, які можуть бути пропущені при автоматизованому тестуванні. Крім того, для проектів з обмеженими ресурсами ручне тестування може бути більш економічно вигідним у короткостроковій перспективі.

Для веб-додатку були розроблені різні тест-кейси, включаючи перевірку функціональності конфігуратора. Наприклад, перевірка роботи конфігуратора передбачає вибір певних компонентів та натискання кнопки "Check Compatibility". Після цього користувач отримує повідомлення про сумісність або несумісність компонентів. Такий підхід дозволяє користувачам упевнитися, що обрані компоненти працюватимуть разом без проблем. Також важливою є перевірка роботи кнопок навігації та кнопки "Вгору". Кнопки навігації мають вести на відповідні сторінки, забезпечуючи зручну та інтуїтивну навігацію по сайту. Кнопка "Вгору" повинна переносити користувача до верхньої частини сторінки, але з'являтися тільки тоді, коли користувач гортатиме сторінку вниз. Таким чином, ручне тестування дозволяє детально перевірити функціональність веб-додатку, виявити та усунути потенційні проблеми перед його випуском.

Автентифікація також є важливою частиною форми, потрібно перевірити, чи працює вона належним чином. Натиснувши кнопку «Check Compatibility», буде отримано повідомлення про помилку сумісності, якщо необхідні компоненти несумісні, також у випадку правильно вибраних компонентів буде виведено повідомлення про успішну сумісність всіх компонентів.

Під час розробки веб-додатку були визначені функціональні вимоги, яких вдалося досягти. Головною метою була забезпечити функціональність конфігуратора, що перевірялося за допомогою тестування його роботи. На рисунку 3.7 відображено процес перевірки функціональності конфігуратора.

Підібрати комплектуючі



The screenshot shows a web-based PC configuration interface. It lists several hardware components, each with a dropdown menu to select a specific model. The components and their selected models are:

- CPU:** Intel Core i9-11900K
- Motherboard:** ASUS ROG Strix Z590-E
- RAM:** Corsair Vengeance LPX 16GB
- GPU:** NVIDIA GeForce RTX 3080
- Storage:** Samsung 970 Evo Plus 1TB
- Power Supply:** Corsair RM750x
- Case:** NZXT H510
- Cooler:** Noctua NH-D15

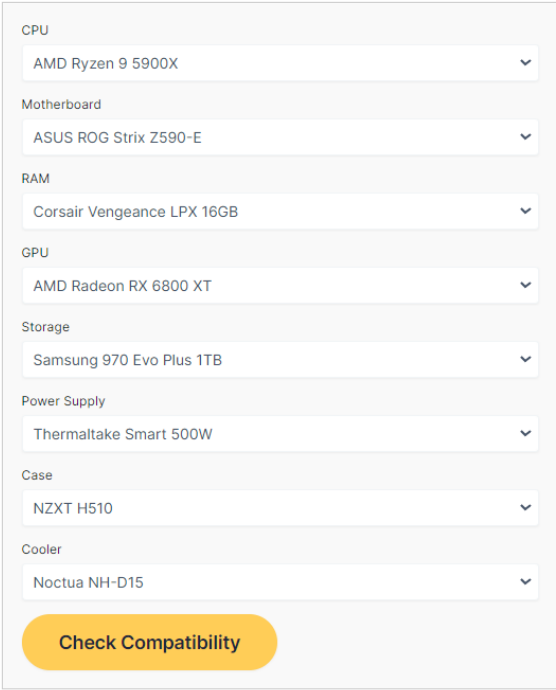
Below the list of components is a yellow button labeled "Check Compatibility". Below the button, a grey message box states: "All parts are compatible!"

Рисунок 3.7 – Перевірка функціональності форми конфігуратора при правильному виборі компонентів

На рисунку 3.7 показано, що при виборі певних компонентів та натисканні на кнопку, форма виводить повідомлення про сумісність усіх обраних компонентів. Для більш глибокої перевірки працездатності конфігуратора необхідно вибирати несумісні між собою компоненти. Для такої перевірки було здійснено заміну процесора з Intel на AMD. Було заздалегідь відомо, що материнська плата не підтримує роботу з процесорами AMD, а кулер не підходить для такого процесора через невідповідність посадкового місця. При такій конфігурації додаток повідомляє користувача про несумісність материнської плати та кулера з процесором, як показано на рисунку 3.8. Це

підтверджує, що конфігуратор успішно виконує свої функції відповідно до поставлених вимог.

Підібрати комплектуючі



The screenshot shows a configuration interface with the following components selected:

- CPU:** AMD Ryzen 9 5900X
- Motherboard:** ASUS ROG Strix Z590-E
- RAM:** Corsair Vengeance LPX 16GB
- GPU:** AMD Radeon RX 6800 XT
- Storage:** Samsung 970 Evo Plus 1TB
- Power Supply:** Thermaltake Smart 500W
- Case:** NZXT H510
- Cooler:** Noctua NH-D15

At the bottom of the configuration list is a yellow button labeled "Check Compatibility". Below the configuration list, a grey box displays the following warnings:

CPU and Motherboard are not compatible.
Cooler and CPU are not compatible.

Рисунок 3.8 – Перевірка функціональності форми конфігуратора при не правильному виборі компонентів

Другою та третьою функціональними вимогами є забезпечення працездатності всіх кнопок, включаючи кнопки навігації, кнопку «вверх» та кнопку форми конфігуратора. Перевірка працездатності кнопок включає впевненість у тому, що вони ведуть на потрібні сторінки та виконують свої функції.

Навігаційні елементи повинні бути інтерактивними та відповідати принципам UI/UX дизайну, забезпечуючи зручну та інтуїтивну навігацію додатком. Навігаційне меню зображене на рисунку 3.9. Для перевірки працездатності кнопок необхідно натискати кожну з них та переконатися, що вони ведуть на правильні сторінки.

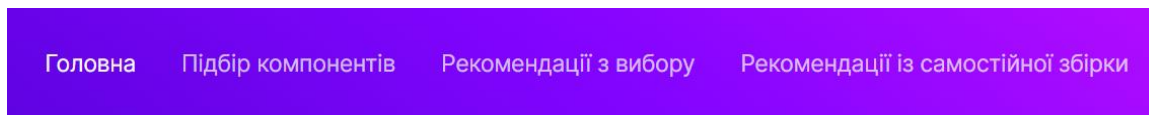


Рисунок 3.9 – Навігаційне меню

Наприклад, при натисканні на кнопку «Підбір компонентів» користувача перенаправляє на сторінку конфігуратора, що підтверджує правильність роботи кнопки. Подібна кнопка на головній сторінці, що називається «Підібрати компоненти», також веде на сторінку конфігуратора, як показано на рисунку 3.10.

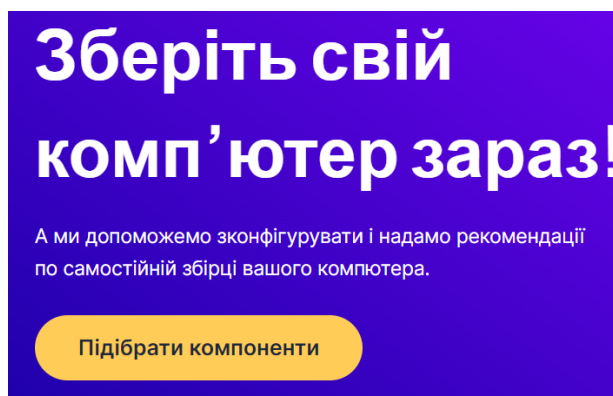


Рисунок 3.10 – Кнопка «Підібрати компоненти»

Кнопки «Рекомендації з вибору» та «Рекомендації із самостійної збірки» ведуть на відповідні статті на сторінці, що відповідає поставленим вимогам.

Кнопка «Вгору», зображена на рисунку 3.11, повинна переносити користувача до верхньої частини сторінки, але з'являтися тільки тоді, коли користувач гортатиме сторінку вниз. Перевірка підтвердила, що цей функціонал реалізований правильно.



Рисунок 3.11 – Кнопка «Вгору»

Покращений веб-додаток має кілька суттєвих переваг над аналогічними продуктами конкурентів:

1. Відсутність рекламної інтеграції: у додатку немає жодних рекламних елементів, що покращує користувацький досвід, зменшує відволікання та прискорює роботу сайту.
2. Оптимізована швидкодія: завдяки оптимізації форми та відсутності зайвих функцій, що не стосуються конфігурування або надання рекомендацій, додаток працює швидше і ефективніше.
3. Свобода вибору магазину: відсутність інтегрованого маркетплейсу дозволяє користувачам самостійно вибирати магазини для покупки обраних компонентів, що дає більше гнучкості та свободи у виборі.
4. Безкорислива основа: додаток базується на безкорисливих засадах, що підвищує довіру користувачів, оскільки немає комерційного інтересу впливати на їхні рішення.

Завдяки цим перевагам, веб-додаток пропонує користувачам більш якісний та зручний інструмент для підбору комп'ютерних компонентів, що робить його конкурентоспроможним на ринку.

3.5 Висновок розділу

В даному розділі було обґрунтовано вибір інструментів розробки веб-додатку, реалізовано форму та логіку додатку, показано етапи їх розробки та етапи наповнення сайту та його дизайну. Також, було проведено ручні тести, перевірено роботу форми та згідно результатам тестування визначено, що вона коректно працює та перевіряє чи правильно працює форма, чи правильно вона визначає сумісність компонентів та виводить інформацію, було перевірено працездатність всіх кнопок та посилань, отже функціональні вимоги були досягнуті. Покращений веб-додаток має переваги над додатками-аналогами.

ВИСНОВКИ

В ході виконання бакалаврської роботи було проведено детальний аналіз найпопулярніших систем управління вмістом (CMS), зокрема їх переваг та недоліків. На основі цього аналізу було обрано та обґрунтовано відповідну CMS для розробки веб-додатку, що надає рекомендації з вибору комп'ютерних комплектуючих.

Було розглянуто різні типи структур веб-сайтів, зокрема деревоподібну структуру, яка була обрана для забезпечення чіткої ієрархії інформації, простоти навігації та зручності оновлення контенту. Ця структура відповідає принципам UI/UX дизайну та гарантує позитивний досвід користувачів. Використання кастомних блоків на сторінках веб-додатку дозволяє динамічно генерувати контент, забезпечуючи гнучкість дизайну та простоту розробки. Покращений веб-додаток має кілька суттєвих переваг над аналогічними продуктами конкурентів. Відсутність рекламної інтеграції в додаток покращує користувацький досвід, зменшує відволікання та прискорює роботу сайту. Завдяки оптимізації форми та відсутності зайвих функцій, що не стосуються конфігурування або надання рекомендацій, додаток працює швидше і ефективніше. Відсутність інтегрованого маркетплейсу дозволяє користувачам самостійно вибирати магазини для покупки обраних компонентів. Додаток базується на безкорисливих засадах, що підвищує довіру користувачів, оскільки немає комерційного інтересу впливати на їхні рішення.

Було обґрунтовано вибір інструментів розробки веб-додатку, реалізовано форму та логіку додатку, показано етапи їх розробки, а також етапи наповнення сайту та його дизайну.

Таким чином, мета роботи — розширення функціональних можливостей Web-додатку для надання рекомендацій з вибору комп'ютерних комплектуючих — була досягнута. Результати тестування підтвердили, що додаток працює коректно та надійно, що підтверджує досягнення поставлених цілей. Порівняння показників нової розробки з показниками аналогів показало, що наш додаток має кращу продуктивність, відсутність реклами та гнучкіші можливості для користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тези доповіді [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20983/17385>
2. Гуртова, Софія. «Системи керування вмістом веб-сайтів.» Інформаційно-комунікаційні технології в освіті 9 (2022).
3. Williams, Brad, David Damstra, and Hal Stern. Professional WordPress: design and development. John Wiley & Sons, 2015.
4. Mariño, S. I., & Alfonzo, P. L. (2018). Web accessibility and CMS. A case study about Joomla and drupal plataforms.
5. Martinez-Caro, Jose-Manuel, et al. «A comparative study of web content management systems.» Information 9.2 (2018): 27.
6. Статистика використання систем управління контентом [Електронний ресурс] – Режим доступу до ресурсу: https://w3techs.com/technologies/overview/content_management
7. Поширення використання CMS по всьому Інтернету [Електронний ресурс] – Режим доступу до ресурсу: <https://trends.builtwith.com/cms/traffic/Entire-Internet>
8. Гук, Н., Диханов, С., & Долотов, І. (2020). Аналіз структури сайту з використанням поняття модулярності. Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки, 99-114.
9. Документація Advanced Custom Fields [Електронний ресурс] – Режим доступу до ресурсу: <https://www.advancedcustomfields.com/resources/>
10. Діренко, І., Олецький, О., & Олецкий, А. (2006). Система управління вмістом веб-ресурсів на основі онтологічно-документного моделювання.
11. Глибовець М. М. Основи комп'ютерних алгоритмів. Київ : Києво-Могилянська академія, 2003. 450 с.
12. Караванова, Т. П., & Костюков, В. П. Програма курсу за вибором «Основи алгоритмізації та програмування».

13. Raes, Filip, et al. «Construction and factorial validation of a short form of the self-compassion scale.» *Clinical psychology & psychotherapy* 18.3 (2011): 250-255.
14. Nixon, Robin. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. « O'Reilly Media, Inc.», 2014.
15. Why Wordpress uses PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://code.tutsplus.com/tutorials/why-wordpress-uses-php--cms-31077>
16. Давиденко, І. С., and Н. П. Бабюк. Аналіз мови програмування JavaScript. Diss. БНТУ, 2022.
17. Meyer, J., Meyer, J., & Corrigan. (2018). *The Essential Guide to HTML5*. Apress.
18. Stewart, Ryan. Brackets 1.0 and Extract for Brackets (Preview) Now Available. Brackets Blog. Adobe Systems.
19. Chaudhary, Mukund, and Ankur Kumar. *PhpStorm Cookbook*. Packt Publishing Ltd, 2014.
20. Christudas, B., & Christudas, B. (2019). *MySQL* (pp. 877-884). Apress.
21. Що таке QA? [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.ithillel.ua/articles/shcho-take-qa-ta-yak-otrymaty-profesiiu-testuvalnyka>
22. Ручне та автоматизоване тестування [Електронний ресурс] – Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/ruchne-ta-avtomatizovane-testuvannya/>

ДОДАТОК А (обов'язковий)
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: WEB-додаток для надання рекомендацій з вибору комп'ютерних компонентів

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Unischek

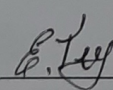
Оригінальність 96,14% Схожість 3,86%

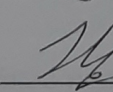
Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Озеранський В.С.

Ознайомлені з повним звітом подібності, який був згенерований системою Unischek щодо роботи.

Автор роботи  Гуменчук Е.С.

Керівник роботи  Малініч І.П.

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА WEB-ДОДАТКУ ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ З ВИБОРУ КОМП'ЮТЕРНИЙ КОМПОНЕНТІВ

Для переходу до конфігуратора необхідно натиснути на кнопку »Підібрати компоненти« яка зображена на рисунку Б.1.

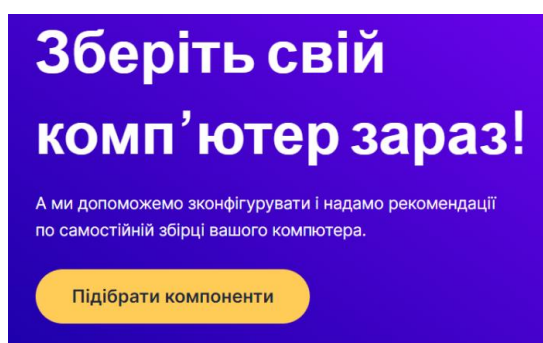


Рисунок Б.1 – Перехід до кнопки

Після цього відкриється вікно конфігуратора, де необхідно з випадаючих списків вибрати необхідні компоненти та натиснути кнопку для перевірки сумісності, після чого буде виведено повідомлення про сумісність компонентів, що зображено на рисунку Б.2.

Підібрати комплектуючі

CPU	Intel Core i9-11900K	▼
Motherboard	ASUS ROG Strix Z590-E	▼
RAM	Corsair Vengeance LPX 16GB	▼
GPU	NVIDIA GeForce RTX 3080	▼
Storage	Samsung 970 Evo Plus 1TB	▼
Power Supply	Corsair RM750x	▼
Case	NZXT H510	▼
Cooler	Noctua NH-D15	▼
Check Compatibility		

All parts are compatible!

Рисунок Б.2 – Вікно конфігуратора

ДОДАТОК В (обов'язковий)

ЛІСТИНГ ПРОГРАМИ

PHP:

```
// Підключення необхідних скриптів і стилів, та створення шорткоду для
// відображення форми підбору комплектуючих
function pc_parts_page() {
    // Підключення необхідних скриптів і стилів
    wp_enqueue_script('jquery');
    wp_enqueue_script('pc-parts-script', get_template_directory_uri() .
'/js/pc-parts.js', array('jquery'), null, true);
    wp_enqueue_style('pc-parts-style', get_template_directory_uri() .
'/css/pc-parts.css');

    // Створення форми
    $output = '<form id=«pc-parts-form»>';

    $components = array(
        'cpu' => 'CPU',
        'motherboard' => 'Motherboard',
        'ram' => 'RAM',
        'gpu' => 'GPU',
        'storage' => 'Storage',
        'psu' => 'Power Supply',
        'case' => 'Case',
        'cooler' => 'Cooler'
    );

    foreach ($components as $key => $label) {
        $output .= '<div class=«component-group»>';
        $output .= '<label for=«' . $key . '»>' . $label . '</label>';
        $output .= '<select id=«' . $key . '» name=«' . $key . '»>';

        // Отримання комплектуючих з бази даних
        $parts = get_pc_parts($key);
        foreach ($parts as $part) {
            $output .= '<option value=«' . esc_attr($part->id) . '»>' .
esc_html($part->name) . '</option>';
        }

        $output .= '</select>';
        $output .= '</div>';
    }

    $output .= '<button type=«submit»>Check Compatibility</button>';
    $output .= '</form>';
    $output .= '<div id=«compatibility-results»></div>';

    return $output;
}
add_shortcode('pc_parts_page', 'pc_parts_page');

// Функція для отримання комплектуючих з бази даних
function get_pc_parts($type) {
    global $wpdb;
```

```

    $results = $wpdb->get_results($wpdb->prepare(«SELECT * FROM
wp_pc_parts WHERE type = %s LIMIT 20», $type));
    return $results;
}

// Підключення вашого скрипта і локалізація змінної ajaxurl
function enqueue_pc_parts_script() {
    // Підключення вашого скрипта
    wp_enqueue_script('pc-parts', get_template_directory_uri() .
'/js/pc-parts.js', array('jquery'), '1.0', true);

    // Локалізація скрипта для передачі PHP змінних у JavaScript
    wp_localize_script('pc-parts', 'pcPartsAjax', array(
        'ajaxurl' => admin_url('admin-ajax.php')
    ));
}
add_action('wp_enqueue_scripts', 'enqueue_pc_parts_script');

// Обробка AJAX-запиту для перевірки сумісності
function check_compatibility() {
    // Отримання даних з запиту
    parse_str($_POST['data'], $formData);

    // Отримання вибраних комплектуючих
    $cpu = isset($formData['cpu']) ? $formData['cpu'] : '';
    $motherboard = isset($formData['motherboard']) ?
$formData['motherboard'] : '';
    $ram = isset($formData['ram']) ? $formData['ram'] : '';
    $gpu = isset($formData['gpu']) ? $formData['gpu'] : '';
    $storage = isset($formData['storage']) ? $formData['storage'] : '';
    $psu = isset($formData['psu']) ? $formData['psu'] : '';
    $case = isset($formData['case']) ? $formData['case'] : '';
    $cooler = isset($formData['cooler']) ? $formData['cooler'] : '';

    // Логіка перевірки сумісності
    $compatible = true;
    $messages = array();

    // Перевірка сумісності CPU і материнської плати
    if ($cpu && $motherboard) {
        global $wpdb;
        $cpu_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'cpu'«,
$cpu));
        $motherboard_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type =
'motherboard'«, $motherboard));

        if ($cpu_data && $motherboard_data) {
            $cpu_data = json_decode($cpu_data, true);
            $motherboard_data = json_decode($motherboard_data, true);

            if (!in_array($cpu_data['socket'], explode(',',
$motherboard_data['socket']))) {
                $compatible = false;
                $messages[] = «CPU and Motherboard are not
compatible.»;
            }
        }
    }
}

```

```

    }
}

// Перевірка сумісності RAM і материнської плати
if ($ram && $motherboard) {
    $ram_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'ram'«,
$ram));
    $motherboard_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type =
'motherboard'«, $motherboard));

    if ($ram_data && $motherboard_data) {
        $ram_data = json_decode($ram_data, true);
        $motherboard_data = json_decode($motherboard_data, true);

        if (!in_array($ram_data['type'], explode(',',
$motherboard_data['memory_type']))) {
            $compatible = false;
            $messages[] = «RAM and Motherboard are not
compatible.»;
        }
    }
}

// Перевірка сумісності GPU і материнської плати
if ($gpu && $motherboard) {
    $gpu_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'gpu'«,
$gpu));
    $motherboard_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type =
'motherboard'«, $motherboard));

    if ($gpu_data && $motherboard_data) {
        $gpu_data = json_decode($gpu_data, true);
        $motherboard_data = json_decode($motherboard_data, true);

        if (!in_array($gpu_data['pci_version'], explode(',',
$motherboard_data['supported_pci_versions']))) {
            $compatible = false;
            $messages[] = «GPU and Motherboard are not compatible.»;
        }
    }
}

// Перевірка сумісності Storage і материнської плати
if ($storage && $motherboard) {
    $storage_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type =
'storage'«, $storage));
    $motherboard_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type =
'motherboard'«, $motherboard));

    if ($storage_data && $motherboard_data) {
        $storage_data = json_decode($storage_data, true);

```

```

        $motherboard_data = json_decode($motherboard_data, true);

        if (!in_array($storage_data['type'], explode(',',
$motherboard_data['storage_interface']))) {
            $compatible = false;
            $messages[] = «Storage and Motherboard are not
compatible.»;
        }
    }
}

// Перевірка сумісності PSU і корпусу
if ($psu && $case) {
    $psu_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'psu'«,
$psu));
    $case_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'case'«,
$case));

    if ($psu_data && $case_data) {
        $psu_data = json_decode($psu_data, true);
        $case_data = json_decode($case_data, true);

        if ($psu_data['form_factor'] !==
$case_data['supported_psu_form_factor']) {
            $compatible = false;
            $messages[] = «PSU and Case are not compatible.»;
        }
    }
}

// Перевірка сумісності Cooler і CPU
if ($cooler && $cpu) {
    $cooler_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'cooler'«,
$cooler));
    $cpu_data = $wpdb->get_var($wpdb->prepare(«SELECT
compatibility_data FROM wp_pc_parts WHERE id = %d AND type = 'cpu'«,
$cpu));

    if ($cooler_data && $cpu_data) {
        $cooler_data = json_decode($cooler_data, true);
        $cpu_data = json_decode($cpu_data, true);

        if (!in_array($cpu_data['socket'], explode(',',
$cooler_data['socket']))) {
            $compatible = false;
            $messages[] = «Cooler and CPU are not compatible.»;
        }
    }
}

if ($compatible) {
    $response = «All parts are compatible!»;
} else {
    $response = implode('<br>', $messages);
}

```

```

}

echo $response;
wp_die();
}
add_action('wp_ajax_check_compatibility', 'check_compatibility');
add_action('wp_ajax_nopriv_check_compatibility',
'check_compatibility');
?>

```

JavaScript:

```

jQuery(document).ready(function($) {
    $('#pc-parts-form').on('submit', function(e) {
        e.preventDefault();

        var formData = $(this).serialize();

        $.ajax({
            url: pcPartsAjax.ajaxurl, // Використовуйте локалізовану
змінну
            type: 'POST',
            data: {
                action: 'check_compatibility',
                data: formData
            },
            success: function(response) {
                $('#compatibility-results').html(response);
            },
            error: function(xhr, status, error) {
                console.error('AJAX Error:', status, error);
            }
        });
    });
});

```

SQL:

```
SELECT * FROM `wp_pc_parts` WHERE 1
```

```

CREATE TABLE wp_pc_parts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    type VARCHAR(50) NOT NULL,
    compatibility_data TEXT
);

```

```

UPDATE wp_pc_parts SET compatibility_data = '{«socket»: «LGA1200»}'
WHERE name = 'Intel Core i9-11900K';
UPDATE wp_pc_parts SET compatibility_data = '{«socket»: «AM4»}' WHERE
name = 'AMD Ryzen 9 5900X';
UPDATE wp_pc_parts SET compatibility_data = '{«socket»: «LGA1200»}'
WHERE name = 'ASUS ROG Strix Z590-E';
UPDATE wp_pc_parts SET compatibility_data = '{«socket»: «AM4»}' WHERE
name = 'MSI MPG B550 Gaming Edge';

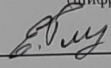
```

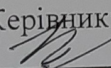
ДОДАТОК Г (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«WEB-ДОДАТОК ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ З ВИБОРУ
КОМП'ЮТЕРНИХ КОМПОНЕНТІВ»

Виконав: студент 4 курсу, групи ЗКН-206
спеціальності 122 – Комп'ютерні науки
(цифр і назва напрямку підготовки, спеціальності)

 Гуменчук Е. С.
(прізвище та ініціали)

Керівник: асистент кафедри КН
 Малініч І. П.
(прізвище та ініціали)

« 07 » 06 2024 р.

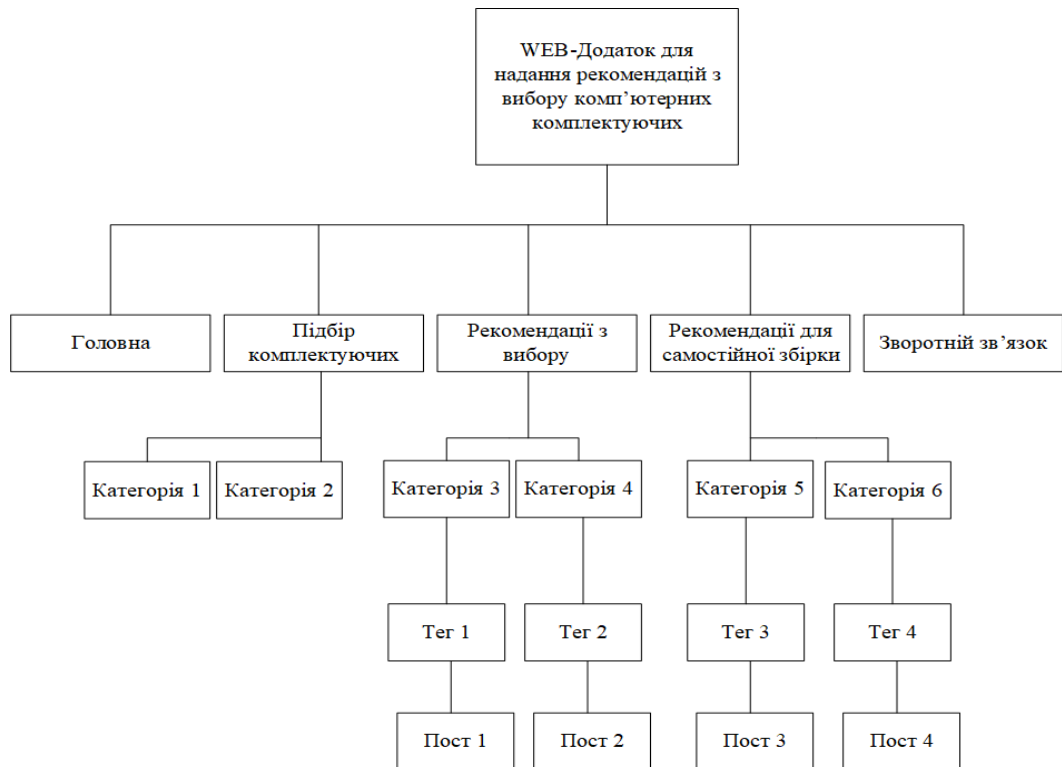


Рисунок Г.1 – Структурна схема веб-додатку

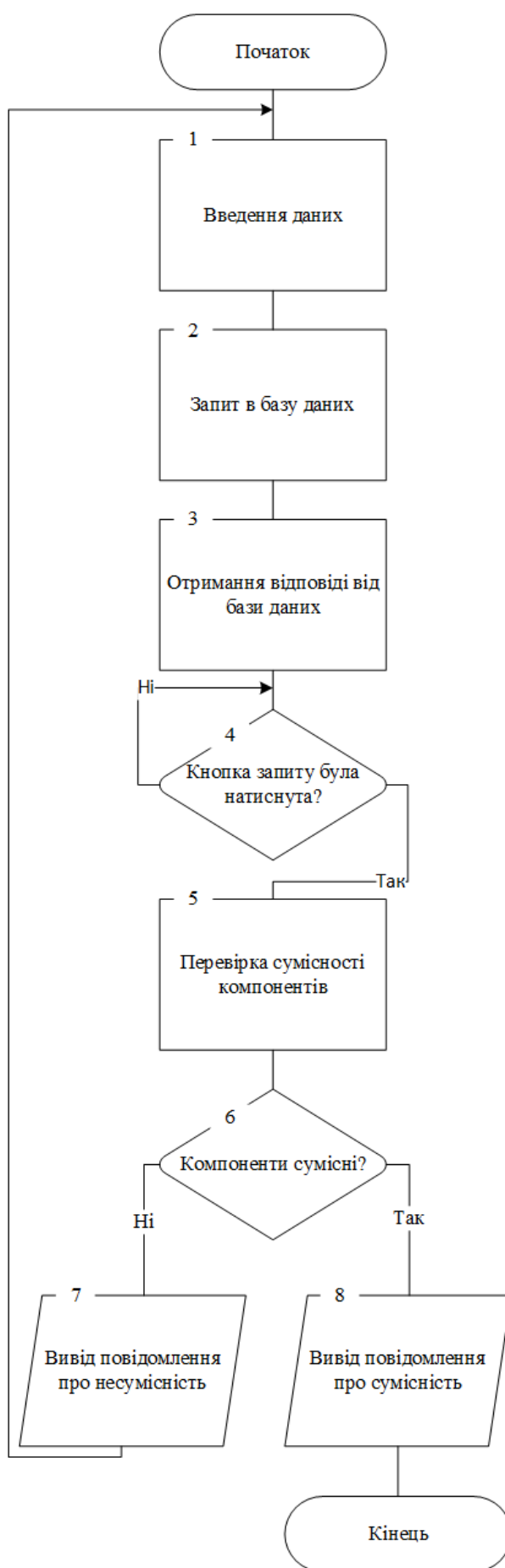


Рисунок Г.2 – Блок-схема алгоритму функціонування форми

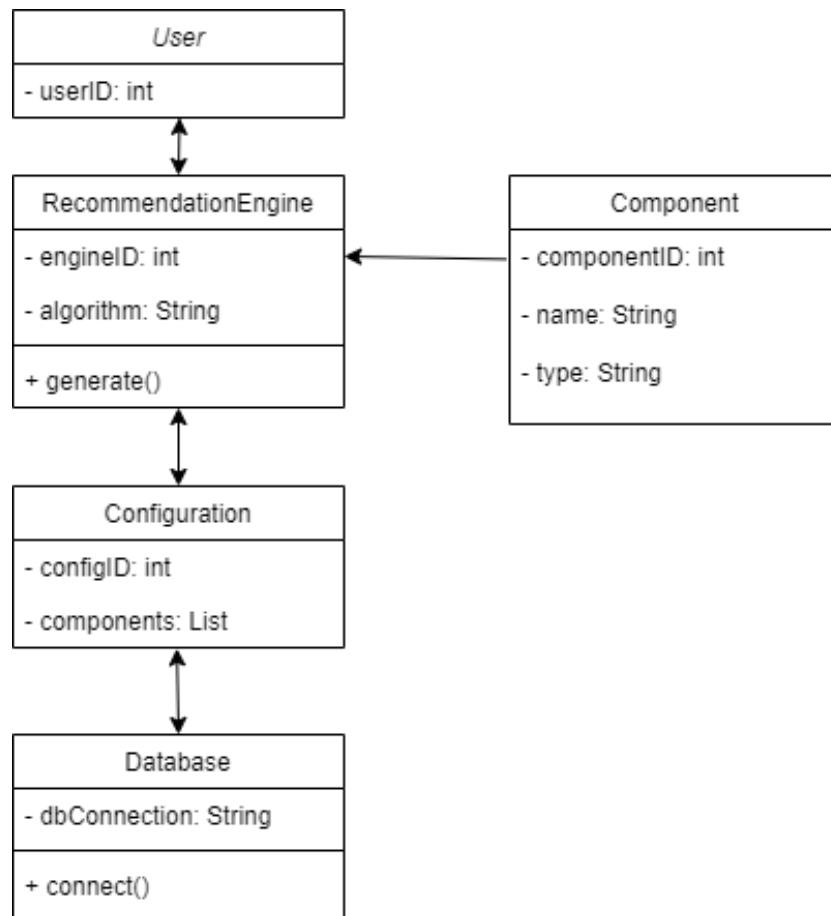


Рисунок Г.3 – UML-діаграма класів

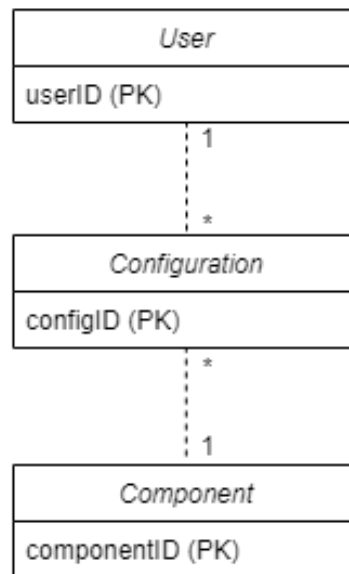


Рисунок Г.4 – ER-діаграма зв'язку між користувачем та базою даних

Підібрати комплектуючі

CPU
Intel Core i9-11900K
Motherboard
ASUS ROG Strix Z590-E
RAM
Corsair Vengeance LPX 16GB
GPU
NVIDIA GeForce RTX 3080
Storage
Samsung 970 Evo Plus 1TB
Power Supply
Corsair RM750x
Case
NZXT H510
Cooler
Noctua NH-D15
Check Compatibility

All parts are compatible!

Рисунок Г.5 – Перевірка функціональності форми конфігуратора при правильному виборі компонентів

Підібрати комплектуючі

CPU
AMD Ryzen 9 5900X
Motherboard
ASUS ROG Strix Z590-E
RAM
Corsair Vengeance LPX 16GB
GPU
AMD Radeon RX 6800 XT
Storage
Samsung 970 Evo Plus 1TB
Power Supply
Thermaltake Smart 500W
Case
NZXT H510
Cooler
Noctua NH-D15
Check Compatibility

CPU and Motherboard are not compatible.
Cooler and CPU are not compatible.

Рисунок Г.6 – Перевірка функціональності форми конфігуратора при не правильному виборі компонентів