

КВАЛІФІКАЦІЙНА РОБОТА

КР.ІПЗ – 22.00.00.000 ІЗ

Група ІПЗс-2017

Онуфрак І.І.

2021

**ЗАКЛАД ВИЩОЇ ОСВІТИ
УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА**

**Факультет суспільних і прикладних наук
Кафедра інформаційних технологій**

на правах рукопису

Онуфрак Іван Іванович

УДК 004.415

Розробка вебресурсу для оптимального підбору компонентів ПК

Спеціальність 121 — «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра

Науковий керівник

к.т.н., доц.,

Ващишак Сергій Петрович

**ЗВО «Університет Короля Данила»
Факультет суспільних і прикладних наук
Кафедра інформаційних технологій**

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ
Завідувач кафедри**

_____ року
« _____ » _____ 202__ року

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

(прізвище, ім'я, по-батькові)

1. Тема роботи

керівник роботи

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «__» _____ 202__ року
№ _____

2. Строк подання студентом роботи

3. Зміст бакалаврської роботи (перелік питань, які потрібно розробити)

4. Дата видачі завдання

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

Сторінка	Опис граф. матеріалу	Сторінка	Опис граф. матеріалу
10	Умовна схема основної конфігурації компонентів ПК	38	CSS для хедеру
13	Шифрувальна машина Enigma		Перша стилізована сторінка застосунку
14	ENIAC - перший електронний ПК	39	Код стилізованих елементів розділення
15	Комп'ютер «Apple I»		Стилізовані елементи розділення
16	Конфігуратор автомобіля BMW	40	Код для стилю списку
17	Конфігуратор для «всього» від Bitrix		Стилізований список елементів
18	Конфігуратор сайту «Фірма АС»	41	Код для стилю характеристик та зображення
19	Приклад частини конфігурації з сайту «Фірма АС»		Стилізований блок елемента
20	Частина конфігурації сайту «GGWP»	42	Код першого елемента сторінки
21	Головна сторінка сайту «EDELWEISS»	43	Код для ціни, магазинів та характеристик
22	Головна сторінка сайту «can.ua»	44	Результат вибору елемента конфігурації
23	Список елементів конфігурації з сайту «can.ua»		Імпортовані дані
31	Загальна структура бази даних	45	Функції для динамічної зміни значень
32	Wireframe головної сторінки		Створення масиву даних
33	Wireframe представлення елементів	46	Присвоєння події кнопці
34	Фінальний Wireframe відображення елементів конфігурації		Дані для графіку
35	Wireframe графіку	47	Код загальної ціни
36	Код для з'єднання з базою даних		Код графіку
37	Код для запиту до перших 4 елементів конфігурації	48	Головний код графіку
	Код для запиту	49	Зображення графіка

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1. ПОНЯТТЯ ТА АНАЛІЗ КОНФІГУРАТОРІВ.....	13
1.1 Історія розвитку ПК.....	13
1.2 Типи та принципи роботи конфігураторів.....	15
1.3 Аналіз існуючих вебконфігураторів для ПК.....	17
1.4 Постановка задачі.....	23
Висновки до розділу 1.....	24
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБЗАСТОСУНКУ.....	26
2.1 Робота з базою даних.....	26
2.3 Розробка Wireframe.....	31
Висновки до розділу 2.....	35
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ.....	36
3.1 Розробка сервера та запитів до бази.....	36
3.2 Стилзація компонентів за допомогою CSS.....	38
3.3 Програмна реалізація компонентів для конфігурації.....	42
3.4 Програмна реалізація виводу графіку ціни.....	44
Висновки до розділу 3.....	49
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	50
4.1 Правила безпечної роботи з комп'ютером.....	50
4.2 Рекомендовані умови для роботи за комп'ютером.....	52
4.3 Розрахунок захисного заземлення.....	52
4.4 Пожежна безпека.....	53
Висновки до розділу 4.....	54

					КР.ПЗ – 22.00.00.000 ПЗ					
					<i>Розробка вебресурсу для оптимального підбору компонентів ПК</i> <i>Пояснювальна записка</i>					
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>						<i>Літ.</i>	<i>Ар.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Онуфрак І.І.</i>								
<i>Перевір.</i>		<i>Ващишак С.П.</i>							6	57
<i>Реценз.</i>		<i>Кузь М.В.</i>						ЗВО «УКД» ПЗс-2017		
<i>Н. Контр.</i>		<i>Зорін В.О.</i>								
<i>Затверд.</i>		<i>Пашкевич О.П.</i>								

ВИСНОВКИ	55
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		7

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- UI – інтерфейс користувача
- База Даних – БД/BD
- ПК – персональний комп'ютер
- EOM – електронно-обчислювальна машина
- ENIAC – Electronic Numerical Integrator and Calculator
- СРСР – Союз Радянських Соціалістичних Республік
- SSD – Solid State Drive
- DPI – Dots Per Inch

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		8

ВСТУП

Актуальність теми. В наш час комп'ютери інтегрувались практично у всі сфери освітньої діяльності людини, починаючи з початкової освіти, закінчуючи вивченням новітніх технологій.

Застосування комп'ютерних технологій полегшує процес освіти в навчальних закладах різного рівня як самих учнів, студентів, так і робочого персоналу. Прикладом цього слугує так необхідне сьогодні СДО, тобто система дистанційної освіти.

Завдяки різноманітності програмного й апаратного забезпечення сьогодні можливе використання всіх потенційних можливостей комп'ютерних технологій. Це дозволяє зберігати та обробляти величезну кількість інформації, використовувати сучасні пакети прикладних програм, працювати з базами даних, здійснювати математичне та дизайнерське моделювання. Беручи до уваги ще й те, що у всі ці пункти можна інтегрувати штучний інтелект, який дасть змогу оптимізувати та покращити ефективність роботи. Сучасні революційні технології в галузі інтелектуальних програм, туманних та хмарних обчислень, виникнення віртуального простору дозволили зберігання практично безмежного обсягу інформації.

Однак, надійна робота комп'ютерної техніки, її швидкодія та обчислювальні можливості залежать, у першу чергу, від конфігурації, правильного підбору, налаштування та узгодження компонентів комп'ютера.

Сучасний ринок пропонує настільки великий вибір різноманітних компонентів для комп'ютера, що легко помилитись обравши вже не актуальний або переоцінений товар. Для людини яка працює в галузі комп'ютерної техніки, постійно слідкує за новинками та в курсі останніх подій в цій сфері не важко вирішити які саме елементи для нього ідеальні.

Але що робити користувачеві який не знайомий з цим? Навіть більше, знайомий тільки з елементарною архітектурою ПК? Як вибрати відеокарту, процесор, материнську плату, накопичувальні диски або навіть елементарні речі, кор-

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		9

пус або периферію? Нище представлена умовна схема (рис. 1) основної конфігурації компонентів ПК та головних компонентів системного блоку.



Рисунок 1 – Умовна схема основної конфігурації компонентів ПК

При тому, що це тільки загальні назви компонентів без назв, моделей та системних характеристик.

Основну, якщо не головну роль в конфігурації грає – системний блок, точніше його наповнення, адже це своєрідне серце комп'ютера.

Вибір обладнання буде складним та довгим, зважаючи на те що кількість товару неймовірно велика і містить в собі моделі, типи та серії компонентів системи під різні навантаження та потреби користувача.

Також немаловажним є узгодження різноманітного обладнання між собою

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		10

за типами з'єднання, драйверами, швидкодією, сумісністю тощо.

Розмаїття номенклатури компонентів ЕОМ та способів їх узгодження і налаштування свідчить про значну потребу у виді сервісу, що даватиме споживачам змогу отримувати оптимальну структуру комп'ютера під свої вимоги.

Тому створення вебресурсу, за допомогою якого як люди, що знайомі з комп'ютерною технікою, так і люди, що практично не мають уявлення про сучасні технології зможуть знайти для себе найкращий варіант конфігурації ПК є досить актуальною задачею.

Правильне вирішення цієї проблеми дасть змогу отримувати оптимальну конфігурацію ПК як з економічного кута зору, так і з кута зору функціональності. Орієнтування буде проводитись як на ціну, так і на головні потреби користувача від системи. Головним завданням для себе автор ставить створення ресурсу орієнтованого більше на дизайн складову ніж на функціональність та на простоту використання ресурсу, щоб навіть дитина зрозуміла що і як потрібно робити. Це не означає що ресурс буде мати скутий функціонал, ні, це означає що дизайн буде виконувати як і свою функцію так і своєрідну механіку конфігурації. Інтегруючи різноманітні «живі» елементи інтерфейсу можна досягнути дивовижного результату навіть коли функціонал програє в деяких речах конкурентам продуманий дизайн з легкістю поверне ресурсу перевагу.

Мета роботи. Розробка вебресурсу для оптимального підбору компонентів ПК під задану конфігурацію.

Об'єкт роботи. Процес підбору компонентів для отримання оптимальної конфігурації ПК.

Предмет роботи. Методика оптимального підбору компонентів ПК.

Завдання роботи:

- пошук та аналіз уже наявних аналогів;
- вибір мови програмування, технологій та інших суміжних програм;
- розробка сучасного та зручного дизайну;
- розробка гнучкого алгоритму збірки компонентів ПК за критеріями;
- проведення тестування продукту.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		11

Методи роботи. Щоб вирішити поставлене завдання були використані відомості про актуальні збірки елементи. Використано мову гіпертекстової розмітки HTML, каскадну таблицю стилів CSS, мову програмування JS та її бібліотеку React.js, базу даних MySQL, середовище програмування Visual Studio Code.

Результати роботи. Результатом кваліфікаційної роботи є вебзастосунок для підбору конфігурації ПК з можливістю аналізу ціни певного елемента збірки, порівняння цін з різних сайтів, інтуїтивним та сучасним дизайном. Розроблений застосунок буде доповнюватись та вдосконалюватись в майбутньому.

Структура роботи. Розділи – 3. Загальний обсяг основної частини – 57. Список використаних джерел містить – 16 позицій.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		12

РОЗДІЛ 1. ПОНЯТТЯ ТА АНАЛІЗ КОНФІГУРАТОРІВ

1.1 Історія розвитку ПК

Ще в початкові часи Китай та Греція розробили для себе аналоги калькуляторів, що було представлено у вигляді мармурної або глиняної дошки з поглибленнями для кульок які слугували цифрами, це був аналог нашої рахівниці, хоча до сьгоднішніх комп'ютерів це має мало відношення, проте це були перші спроби людей полегшити обчислення, що і є основною метою персонального ПК.

За 103 роки комп'ютерні технології розкинулись з шаленим темпом, все починається з бажання розшифрувати німецькі шифри, що стало приводом створення «Enigma» (рис. 1.1) — першої шифрувальної машини, а закінчилось сучасними ПК які можуть обробляти терабайти інформації всього за секунди.



Рисунок 1.1 – Шифрувальна машина Enigma

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		13

Історія показує, що бажання людей полегшити собі життя призводять до неймовірних результатів, сьогодні для студента програміста написати програму яка буде виконувати 5000 тисяч операцій додавання і 300 віднімання в секунду це курсова робота, а в 1946 році це був комп'ютер ENIAC (рис.1.2), який був тридцяти тонною «шафою» та займає площу середнього футбольного поля. Для роботи обчислень ENIAC використовував лампи в кількості 18 тисяч, що робило його роботу малоефективною за внаслідок того, що кожну годину 70-90 лам виходили з ладу, що припиняє роботу обчислень.

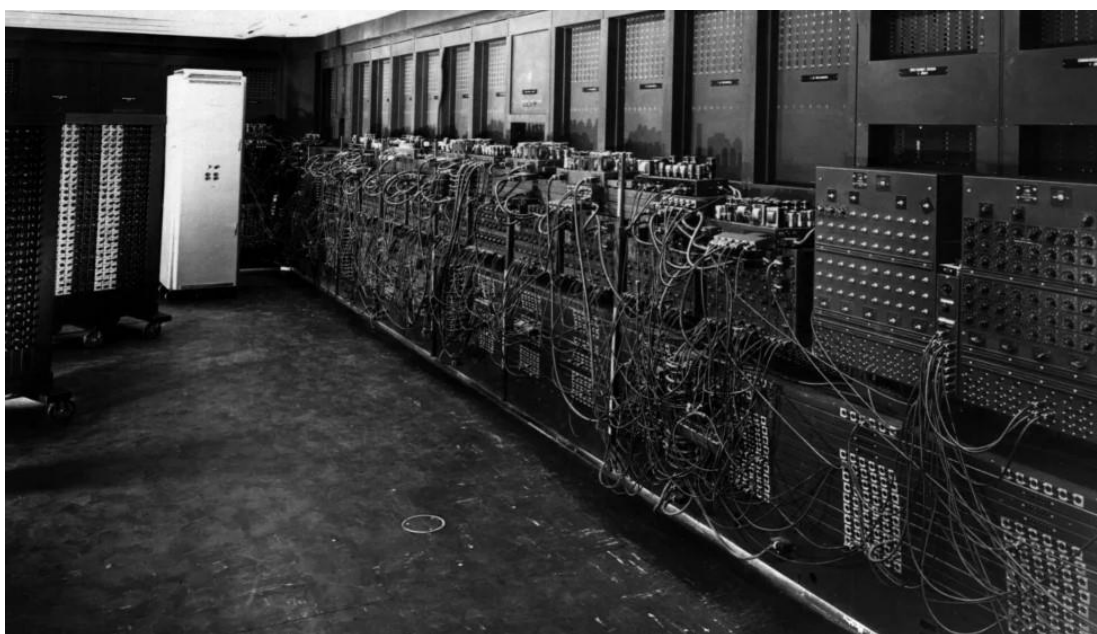


Рисунок 1.2 – ENIAC перший електронний ПК

Не дивлячись на те, що перший ПК хоча і не реалізований у «залізі», але з'явився ще в СРСР завдяки інженеру Арсенію Горохову, який в 1973 році отримав авторське свідоцтво №383005, в якому описав «програмний пристрій» та на жаль фінансування проєкт не здобув, так і залишившись просто авторським свідоцтвом №383005. По всьому світі почались своєрідні спроби створення персонального комп'ютера.

Ще до Горохова, Конрад Цузе у 1936 році сконструював апарат, що працює в двійковій системі числення, названий «Zuse». В 1976 році, Стівом Возняком був спроектований «Apple I» (рис. 1.3) з яким пов'язано декілька цікавих

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		14

фактів. Насамперед кількість випущених моделей не перевищувала і двох сотень, а ціна складала 666,66 долара. Ціну встановив сам творець, така ціна вибрана тому, що він любив повторювані цифри. Не дивлячись на те, що в ті часи після покупки «Apple I» потрібно було купити ще корпус, екран, клавіатуру та живлення, в наш час аукціонний дім «Christie's» продав одну з таких моделей у 2014 році за 905 тисяч доларів, що дорівнює приблизно 25 мільйонів гривень на сьогоднішній курс.

Доволі дивно спостерігати як в наш час, коли покупки одного SSD вистачило б на весь потенціал пам'яті всіх «Apple I» разом взятих і ще 25 мільйонів залишилося б. Але все ж таки я думаю «Apple I» купили для збереження історії, а не для практичних цілей.



Рисунок 1.3 – Комп'ютер «Apple I»

1.2 Типи та принципи роботи конфігураторів

Хорошим конфігуратором вважається той в якого є велика база в компонентах, функціонал який дозволяє або купити, або хоча б переглянути магазини де ці компоненти можна купити, також перевагою буде аналіз цін загальної збірки або кожного компонента окремо, що дозволить зрозуміти в якому місці, якщо розглядати збірку ПК, ваша система слабша, або навпаки занадто дорога або

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		15

просто компоненти не підходять один одному.

При сучасному ринку, коли конкуренція в практично кожній сфері шалена, важко вибрати щось краще та відповідне під ваші конкретні вимоги, на вирішення цієї проблеми якраз і створено конфігуратори, погодьтесь краще онлайн зібрати мотор для машини та зрозуміти що він для вас не достатньо потужний ніж зрозуміти це безпосередньо під час збірки в гаражі.

Види конфігураторів, як напевно кожного товару існують два, платний та безплатний. Перший відповідно має кращий функціонал, більшу базу, можливості. Другий легко доступний шляхом своєї обмеженості в діях, хоча якщо розглядати саме ПК конфігурацію, то на сьогоднішньому ринку доволі важко виділятися чимось унікальним.

Доволі популярними є конфігуратори автомобіля (рис. 1.4) в яких можна візуально побачити машину в 3D варіанті, покрутити її, зазирнути в салон, змінити колір, поміняти диски.

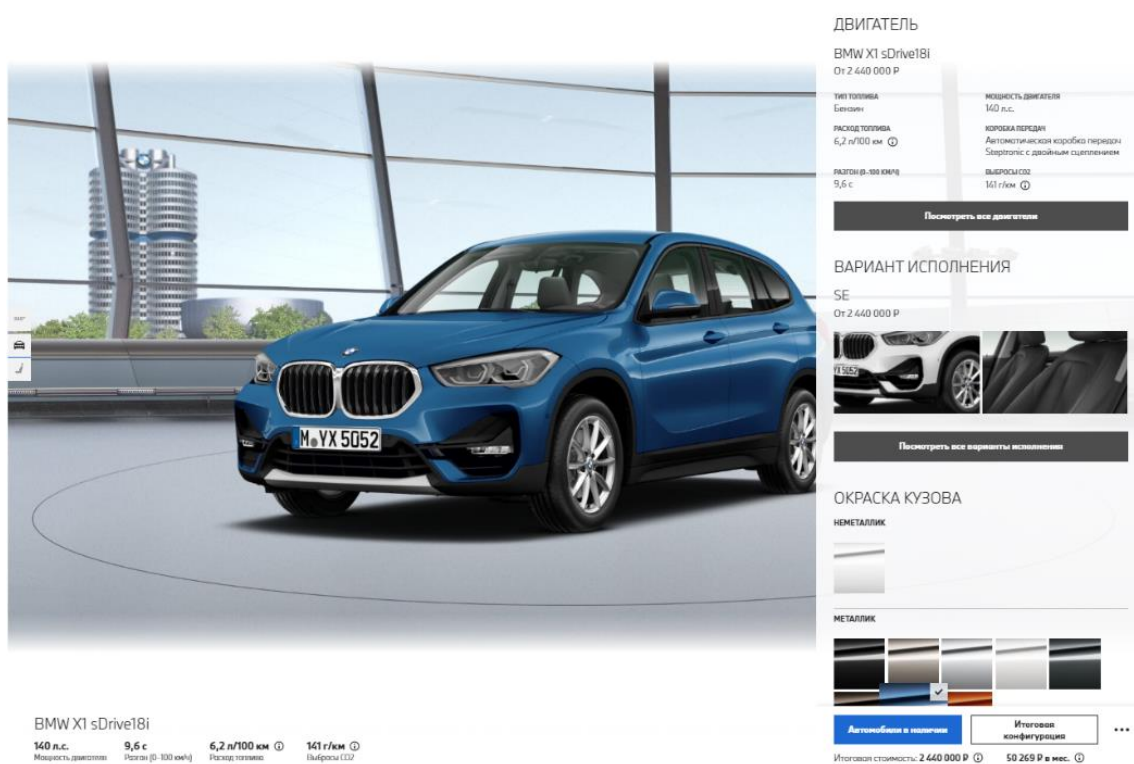


Рисунок 1.4 – Конфігуратор автомобіля BMW

Також існують конфігуратори «для всього» (рис. 1.5), які хоч і описують

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		16

свої можливості як конфігуратори, насправді є звичайними фільтрами.

Необхідно розуміти, що фільтр це алгоритм який видає результат по заданому критерію, а конфігуратор це збірка елементів чогось одного в єдиний комплект. Виходячи з цього, на мою думку, такі конфігуратори хоч і мають неймовірну базу та необмежений функціонал, але вони не мають головного – індивідуальності та сфокусованості на чомусь одному, можливо краще більше ніж якісніше, але якщо діло стосується конфігураторів, краще обирати вузько направлені речі, щоб могли максимально точно та ефективно підібрати та аналізувати елементи.

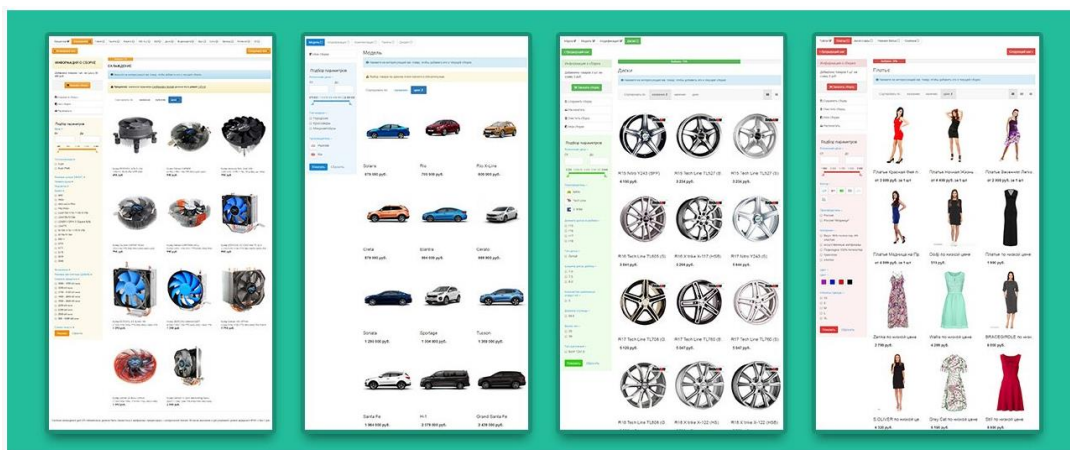


Рисунок 1.5 – Конфігуратор для «всього» від Vitrix

1.3 Аналіз існуючих вебконфігураторів для ПК

В наш час в інтернеті можна знайти будь-що та використати будь-який ресурс для різноманітних задач. Використати онлайн калькулятор для обчислення, магазин для покупки товару, сайт новин, щоб бути в курсі актуальних новин та стану світу загалом.

Комп'ютери як і ІТ розвиваються швидко та з шаленими темпами, тому щоб розуміти суть та бути в тренді з новинками ПК галузі необхідно стежити за новинами певних ресурсів або інтернет журналів. Або як компроміс скористатись онлайн сервісами які допоможуть з комплектацією вашої системи, дадуть розуміння того чи іншого елементу та аналіз ринку загалом.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		17

Конфігураторами користуються люди які хочуть об'легшити собі життя, та підібрати собі щось під свої потреби та фінансові можливості.

Розглянемо деякі з онлайн конфігураторів та проаналізуємо переваги та недоліки кожного з них.

Сайт Фірма АС[1].

Фірма АС – це інтернет-магазин в якому можна придбати товари від мобільних аксесуарів до кондиціонерів. Конфігуратор ПК (рис. 1.6) являє собою тільки додаткову функцію.

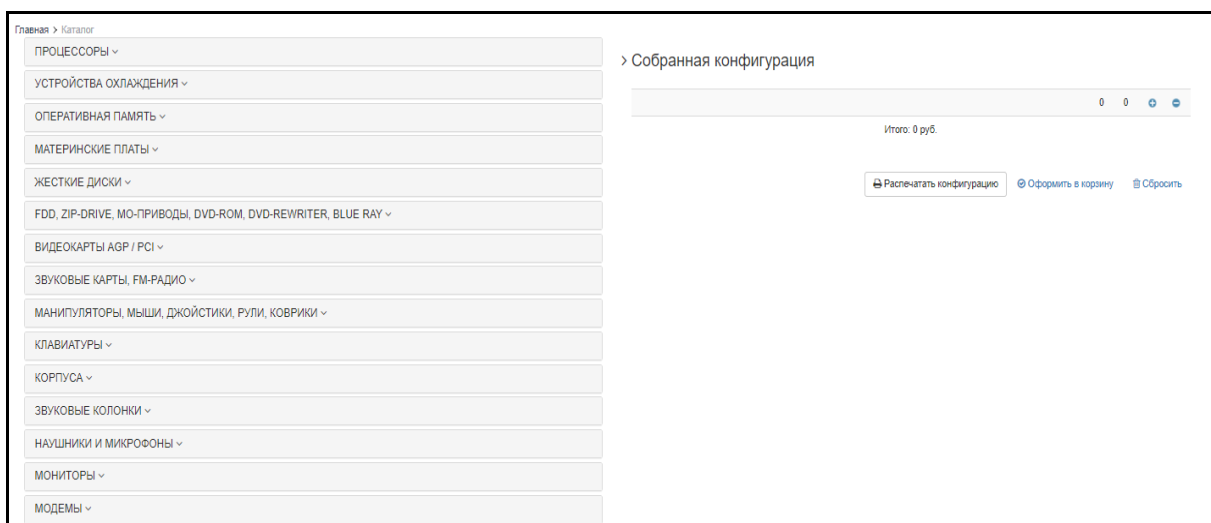


Рисунок 1.6 – Конфігуратор сайту «Фірма АС»

На сайті також є статті про новинки комп'ютерної техніки, та відповідна документація.

Переваги сайту аналога:

- велика кількість елементів для конфігурації;
- пошук та фільтрація елементів;
- можливість перегляду загальної ціни та ціни конкретного елемента;
- друк конфігурації.

Недоліки:

- незручність користування через те, що меню компонента відкривається та закриває основне;

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		18

- не структурований короткий опис елемента;
- замалий шрифт ціни та сторінки в цілому;
- попередній огляд товару здійснюється на новій вкладці, що не зручно;
- малі фото в попередньому перегляді (рис. 1.7);
- відсутність збереження конфігурації.

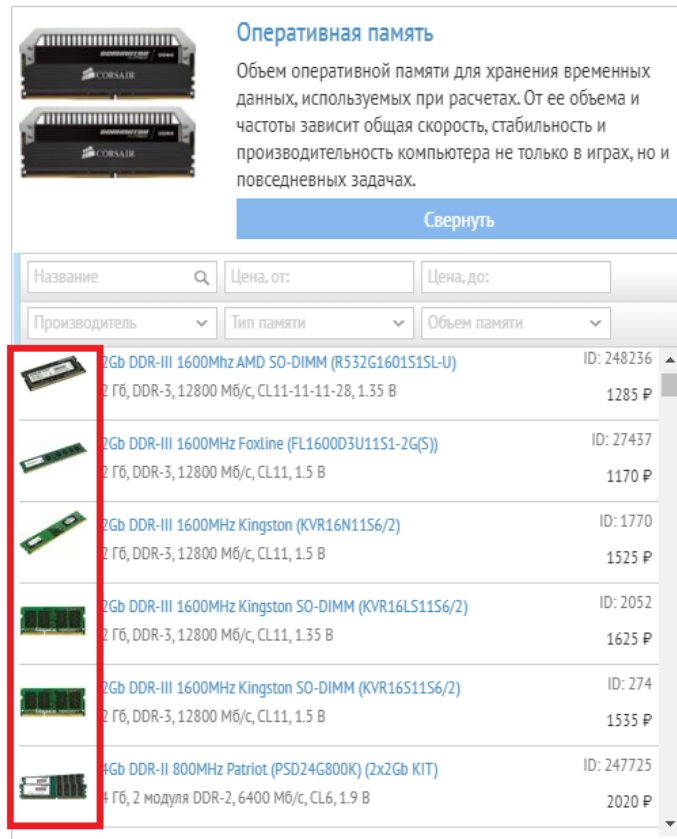


Рисунок 1.7 – Приклад частини конфігурації з сайту «Фірма АС»

Сайт Power GGWP [2].

Цей сайт спеціалізується на продажі та створенні власних конфігурацій для ПК видає гарантійний талон та всю необхідну документацію. Конфігуратор на сайті розроблено у вигляді списку (рис. 1.8).

Переваги сайту-аналога:

- приємний дизайн та підібрані кольори;
- великі фото елементів конфігурації;

- можливість перегляду загальної ціни конфігурації;
- сервісна підтримка за телефоном та чатом;
- ліцензія Windows 10 в подарунок.

Недоліки:

- відсутність перегляду ціни кожного елемента в фінальній конфігурації;
- малий асортимент елементів;
- немає можливості переглянути елемент цілком;
- не можна зберегти конфігурацію;
- відсутність фільтрування за ціною;
- відсутність порівняння ціни.

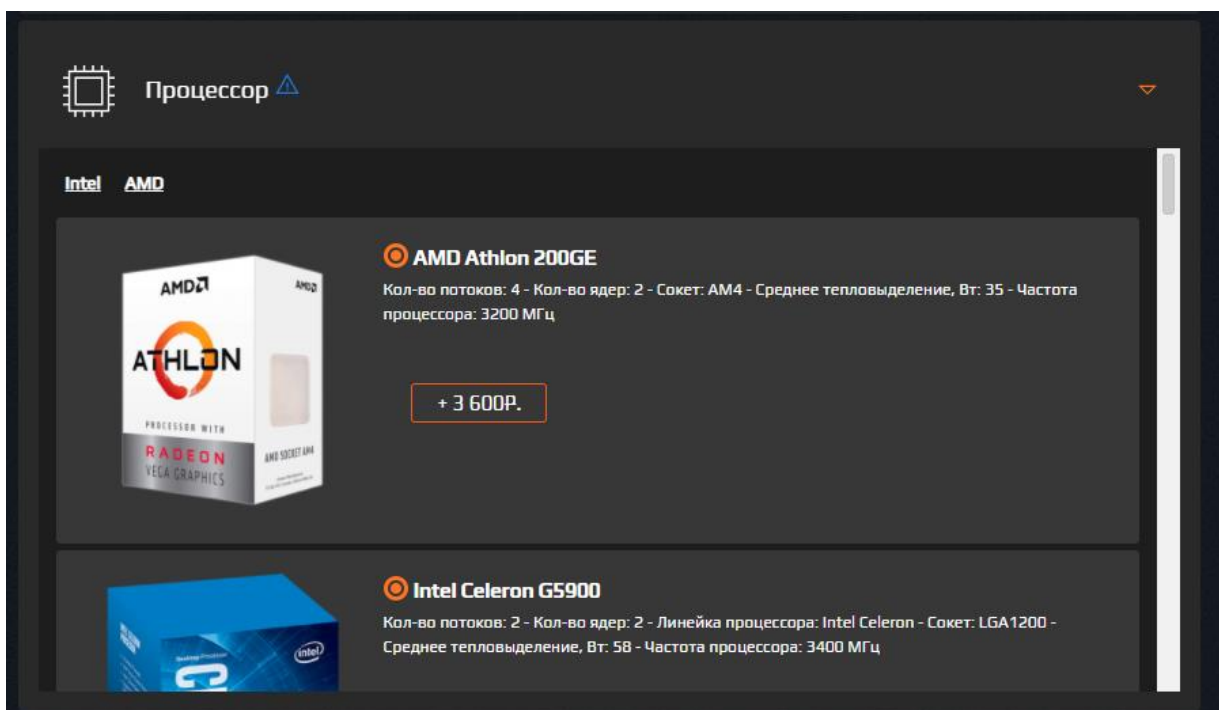


Рисунок 1.8 – Частина конфігурації сайту «GGWP»

Сайт EDELWEISS [3].

Сайт повністю присвячений конфігурації під різні задачі (рис. 1.9) а точніше:

- недорогі збірки;
- ігрові;

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		20

- найпотужніші;
- ПК на базі AMD;
- ПК під гру.

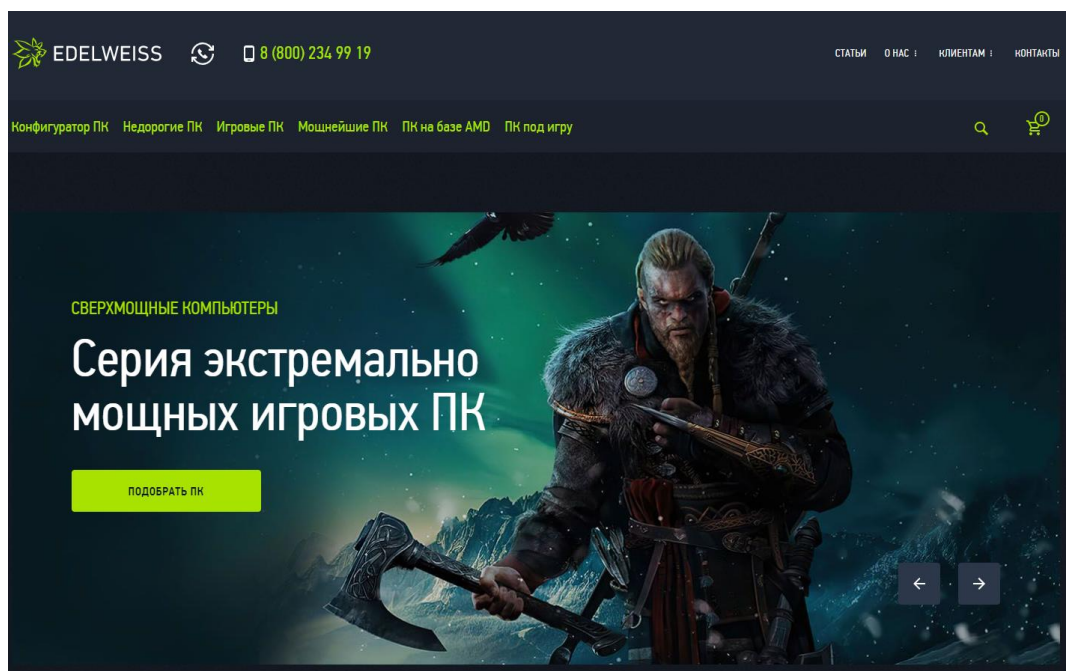


Рисунок 1.9 – Головна сторінка сайту «EDELWEISS»

Переваги сайту-аналога:

- приємний дизайн та підібрані кольори;
- великий вибір конфігурацій під різні потреби;
- великий обсяг елементів;
- можливість переглянути загальну ціну конфігурації.

Недоліки:

- відсутність перегляду ціни кожного елемента в фінальній конфігурації;
- реалізація вибору елементів, прапорцем не зручна;
- немає можливості переглянути елемент цілком;
- відсутність фільтрування за ціною;
- відсутність порівняння ціни;
- немає коротких характеристик елементу.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		21

Сайт can.ua [4].

Сайт являє собою інтернет-магазин (рис. 1.10) різноманітної комп'ютерної техніки, аксесуарів, телефонів та інших електро пристроїв.

Головна сторінка сайту занадто заповнена товаром, що не дає змогу сфокусуватись на чомусь конкретному.

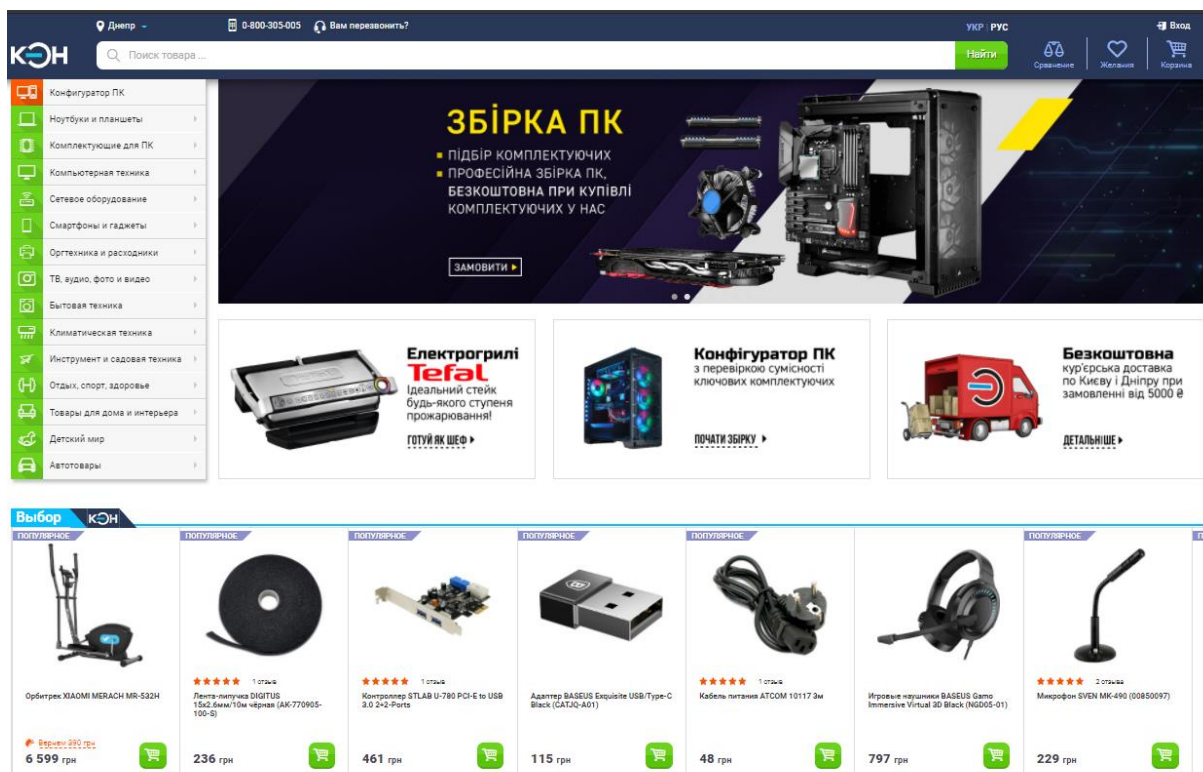


Рисунок 1.10 – Головна сторінка сайту «can.ua»

На сайті конфігуратор це частина функціоналу, реалізований як окрема сторінка зі списком компонентів та кнопками (рис. 1.11), перехід безпосередньо до елементів відбувається через нове вікно та завантаження сторінки що не зручно та при тривалому використанні сайту дратує.

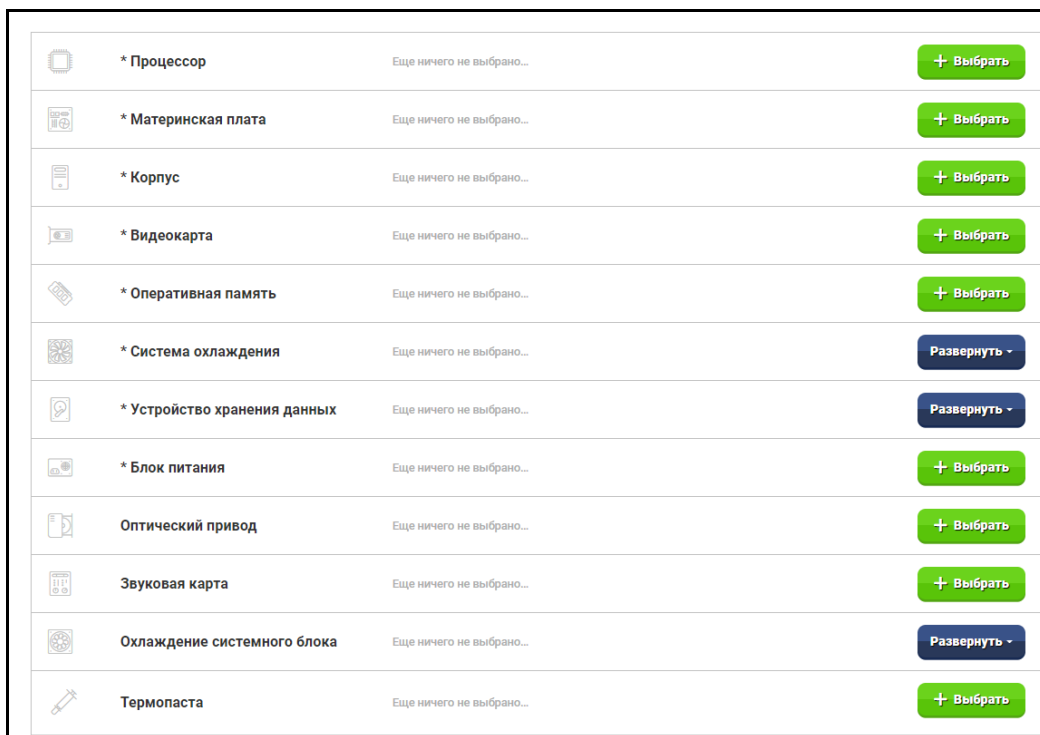
Переваги сайту–аналога:

- великий асортимент елементів для конфігурації;
- добре структурований опис елементу;
- перегляд ціни конкретного елементу;
- можливість переглянути загальну ціну конфігурації.

									Арк.
									22
Змн.	Арк.	№ докум.	Підп.	Дата	КР.ІІЗ – 22.00.000 ІІЗ				

Недоліки:

- відсутність перегляду ціни кожного елемента в фінальній конфігурації;
- для перегляду списку елементів, відкривається нове вікно що не зручно.















	* Процессор	Еще ничего не выбрано...	+ Выбрать
	* Материнская плата	Еще ничего не выбрано...	+ Выбрать
	* Корпус	Еще ничего не выбрано...	+ Выбрать
	* Видеокарта	Еще ничего не выбрано...	+ Выбрать
	* Оперативная память	Еще ничего не выбрано...	+ Выбрать
	* Система охлаждения	Еще ничего не выбрано...	Развернуть ▾
	* Устройство хранения данных	Еще ничего не выбрано...	Развернуть ▾
	* Блок питания	Еще ничего не выбрано...	+ Выбрать
	Оптический привод	Еще ничего не выбрано...	+ Выбрать
	Звуковая карта	Еще ничего не выбрано...	+ Выбрать
	Охлаждение системного блока	Еще ничего не выбрано...	Развернуть ▾
	Термопаста	Еще ничего не выбрано...	+ Выбрать

Рисунок 1.11 – Список елементів конфігурації з сайту «can.ua»

З наведеного вище можна зробити висновок, що жоден з конфігураторів хоча і має свої переваги, будь то приємний дизайн або велика база товару, все ж таки мають і недоліки, які розроблювальний проєкт повинен вирішити. Для кращого розуміння конкретних цілей та вимог до застосунку, які сформувались в результаті аналізу попередніх сайтів, необхідно створити список цілей та основні задачі.

1.4 Постановка задачі

В першому розділі було розглянуто історію персонального комп'ютера, де-

які з перших прототипів. Проведено аналіз конфігураторів та їх видів. Проведено перегляд сайтів аналогів застосунку, описано переваги та недоліки. Проаналізувавши аналоги було поставлено мету створити конфігуратор який буде без описаних недоліків і буде покращувати наявні переваги. Проаналізувавши всі аналоги та створивши необхідні критерії для конфігуратора, було поставлено до виконання наступні цілі:

1. Створення сучасного та зрозумілого дизайну;
2. Достатньо великі фото компонентів конфігурації;
3. Опис основних характеристик компонентів;
4. Функціонал збереження, друку конфігурації;
5. Можливість перегляду загальної ціни та ціни конкретного елемента збірки;
6. Наявність порівняння цін з іншими магазинами;

Кожен з елементів конфігурації буде представлений у вигляді списку, після його вибору користувач побачить певні характеристики, орієнтовну ціну та зможе переглянути пропозиції магазинів. Після вибору всіх представлених компонентів конфігурації, користувач зможе провести аналіз загальної ціни, та окремо % від ціни кожного елемента. Реалізація пошуку та фільтрів також розглядається в перспективі проекту, але для початку потрібно зосередитись на вже визначених цілях та їх програмній реалізації.

Висновки до розділу 1

В розділі було досліджено історію перших спроб реалізувати комп'ютер, як вдалих, так і не дуже. Проведено аналіз конфігураторів за призначенням та функціональністю, наведено деякі аналоги.

Розглянуто схожі за тематикою до проекту, конфігуратори та проведено аналіз їх переваг та недоліків.

З аналізу аналогів також було описано конкретні задачі які мають вдосконалити продемонстровані переваги аналогів та виправити недоліки. Описано ос-

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		24

новний функціонал який повинен бути розроблений в ході виконання кваліфікаційної роботи та розроблено умовний концепт проєкту.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		25

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБЗАСТОСУНКУ

2.1 Робота з базою даних

Для зберігання даних та наступного їх використання було спроектовано сервер та базу даних. Для цього було використано середовище MySQL Workbench та мову структурованих запитів SQL.

Для початку створимо таблиці тільки для деяких елементів конфігурації комп'ютера, а саме:

1. Процесор;
2. Материнська Плата;
3. Відеокарта;
4. SSD;
5. Монітор;
6. Клавіатура;
7. Миша;
8. Навушники.

Створимо таблицю Процесор з назвою «processors», з полями:

- idProcessors (порядковий номер запису);
- Name_Processors (назва процесору);
- Price_Processors (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини) ;
- NumberKernels (кількість ядер);
- ProcessorFrequency (частота процесора);
- MaxMemory (максимальний об'єм пам'яті).

Код для створення таблиці «processors»:

```
CREATE TABLE `pc_config`.`processors` (  
  `idProcessors` INT NOT NULL AUTO_INCREMENT,
```

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		26

```

`Name_Processors` VARCHAR(45) NULL DEFAULT NULL,
`Price_Processors` FLOAT NULL DEFAULT NULL,
`Img_src` VARCHAR(350) NULL DEFAULT NULL,
`LinkForPurchase` VARCHAR(350) NULL DEFAULT NULL,
`NumberKernels` INT NULL DEFAULT NULL,
`ProcessorFrequency` FLOAT NULL DEFAULT NULL,
`MaxMemory` INT NULL DEFAULT NULL,
PRIMARY KEY (`idProcessors`);

```

Таблиця Материнська Плата з іменем «motherboards» та полями:

- idmotherboards (порядковий номер запису);
- Name_Motherboards (назва материнської плати);
- Price_Motherboards (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини) ;
- NumberMS (кількість слотів пам'яті);
- NumberC (кількість каналів);
- MaxMemory (максимальний об'єм пам'яті).

Код для створення таблиці «motherboards»:

```

CREATE TABLE `pc_config`.`«motherboards»` (
  `idmotherboards` INT NOT NULL AUTO_INCREMENT,
  `Name_Motherboards` VARCHAR(45) NULL DEFAULT NULL,
  `Price_Motherboards` INT NULL DEFAULT NULL,
  `Img_src` VARCHAR(350) NULL DEFAULT NULL,
  `LinkForPurchase` VARCHAR(45) NULL DEFAULT NULL,
  `NumberMS` INT NULL DEFAULT NULL,
  `NumberC` INT NULL DEFAULT NULL,
  `MaxMemory` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idmotherboards`));

```

Таблиця Відеокарта з іменем «videocard» та полями:

- idvideocard (порядковий номер запису);
- Name_Videocard (назва відеокарти);
- Price_Videocard (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини) ;
- MemoryCapacity (об'єм пам'яті);
- ProcessorFamily (сімейство процесора);

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		27

- MaxResolution (максимальне розширення).

Код для створення таблиці «videocard»:

```
CREATE TABLE `pc_config`.`videocard` (
  `idvideocard` INT NOT NULL AUTO_INCREMENT,
  `idvideocard_copy1` VARCHAR(45) NULL DEFAULT NULL,
  `Price_Videocard` INT NULL DEFAULT NULL,
  `Img_src` VARCHAR(345) NULL DEFAULT NULL,
  `LinkForPurchase` VARCHAR(45) NULL DEFAULT NULL,
  `MemoryCapacity` INT NULL DEFAULT NULL,
  `ProcessorFamily` VARCHAR(45) NULL DEFAULT NULL,
  `MaxResolution` VARCHAR(100) NULL DEFAULT NULL,
  PRIMARY KEY (`idvideocard`));
```

Таблиця SSD з іменем «ssd» та полями:

- idSSD (порядковий номер запису);
- Name_SSD (назва SSD);
- Price_SSD (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини) ;
- MemoryCapacity (об'єм пам'яті);
- ReadingS (швидкість зчитування);
- RecordingS (швидкість запису).

Код для створення таблиці «ssd»:

```
CREATE TABLE `pc_config`.`ssd` (
  `idSSD` INT NOT NULL,
  `Name_SSD` VARCHAR(45) NULL DEFAULT NULL,
  `Price_SSD` INT NULL DEFAULT NULL,
  `Img_src` VARCHAR(345) NULL DEFAULT NULL,
  `LinkForPurchase` VARCHAR(45) NULL DEFAULT NULL,
  `MemoryCapacity` VARCHAR(50) NULL DEFAULT NULL,
  `ReadingS` INT NULL DEFAULT NULL,
  `RecordingS` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idSSD`));
```

Таблиця Монітор з іменем «monitor» та полями:

- idmonitor (порядковий номер запису);
- Name_Monitor (назва монітору);
- Price_Monitor (Ціна);

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		28

- `Img_src` (шлях до зображення процесора);
- `LinkForPurchase` (посилання на магазини) ;
- `Diagonal` (діагональ);
- `ScreenResolution` (розширення екрану);
- `RatioParties` (відношення сторін).

Код для створення таблиці «monitor»:

```
CREATE TABLE `pc_config`.`monitor` (
  `idmonitor` INT NOT NULL,
  `Name_Monitor` VARCHAR(45) NULL DEFAULT NULL,
  `Price_Monitor` INT NULL DEFAULT NULL,
  `Img_src` VARCHAR(445) NULL DEFAULT NULL,
  `LinkForPurchase` VARCHAR(900) NULL DEFAULT NULL,
  `Diagonal` FLOAT NULL DEFAULT NULL,
  `ScreenResolution` VARCHAR(45) NULL DEFAULT NULL,
  `RatioParties` VARCHAR(55) NULL DEFAULT NULL,
  PRIMARY KEY (`idmonitor`));
```

Таблиця Клавіатура з іменем «keyboard» та полями:

- `idkeyboard` (порядковий номер запису);
- `Name_Keyboard` (назва клавіатури);
- `Price_Keyboard` (Ціна);
- `Img_src` (шлях до зображення процесора);
- `LinkForPurchase` (посилання на магазини) ;
- `Type` (тип);
- `Conect` (з'єднання);
- `ResourceKey` (ресурс клавіш).

Код для створення таблиці «keyboard»:

```
CREATE TABLE `pc_config`.`keyboard` (
  `idkeyboard` INT NOT NULL,
  `Name_Keyboard` VARCHAR(45) NULL DEFAULT NULL,
  `Price_Keyboard` INT NULL DEFAULT NULL,
  `Img_src` VARCHAR(900) NULL DEFAULT NULL,
  `LinkForPurchase` VARCHAR(900) NULL DEFAULT NULL,
  `Type` VARCHAR(45) NULL DEFAULT NULL,
  `Conect` VARCHAR(45) NULL DEFAULT NULL,
  `ResourceKey` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idkeyboard`));
```

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		29

Таблиця Миша з іменем «mouse» та полями:

- idmouse (порядковий номер запису);
- Name_Mouse (назва миші);
- Price_Mouse (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини);
- Conect (з'єднання);
- DPI (DPI);
- NumberB (кількість кнопок).

Код для створення таблиці «mouse»:

```
CREATE TABLE `pc_config`.`mouse` (  
  `idmouse` INT NOT NULL,  
  `Name_Mouse` VARCHAR(45) NULL DEFAULT NULL,  
  `Price_Mouse` INT NULL DEFAULT NULL,  
  `Img_src` VARCHAR(405) NULL DEFAULT NULL,  
  `LinkForPurchase` VARCHAR(905) NULL DEFAULT NULL,  
  `Conect` VARCHAR(45) NULL DEFAULT NULL,  
  `DPI` VARCHAR(45) NULL DEFAULT NULL,  
  `NumberB` INT NULL DEFAULT NULL,  
  PRIMARY KEY (`idmouse`));
```

Таблиця Навушники з іменем «headphones» та полями:

- idheadphones (порядковий номер запису);
- Name_Headphones (назва навушників);
- Price_Headphones (Ціна);
- Img_src (шлях до зображення процесора);
- LinkForPurchase (посилання на магазини);
- Type (тип);
- Conect (з'єднання);
- LengthCable (довжина проводу).

Код для створення таблиці «headphones»:

```
CREATE TABLE `pc_config`.`headphones` (  
  `idheadphones` INT NOT NULL,  
  `Name_Headphones` VARCHAR(65) NULL DEFAULT NULL,  
  `Price_Headphones` INT NULL DEFAULT NULL,
```

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		30

```

`Img_src` VARCHAR(405) NULL DEFAULT NULL,
`LinkForPurchase` VARCHAR(945) NULL DEFAULT NULL,
`Type` VARCHAR(45) NULL DEFAULT NULL,
`Conect` VARCHAR(45) NULL DEFAULT NULL,
`LengthCable` FLOAT NULL DEFAULT NULL,
PRIMARY KEY (`idheadphons`));

```

Після проектування та створення всіх необхідних таблиць (рис. 2.1) з коректними полями та типами, можна переходити до наступного етапу розробки ресурсу проектування прототипу в якому було представлено чорно-білий варіант застосунку з можливим функціоналом та умовним розташуванням необхідних елементів для застосунку.

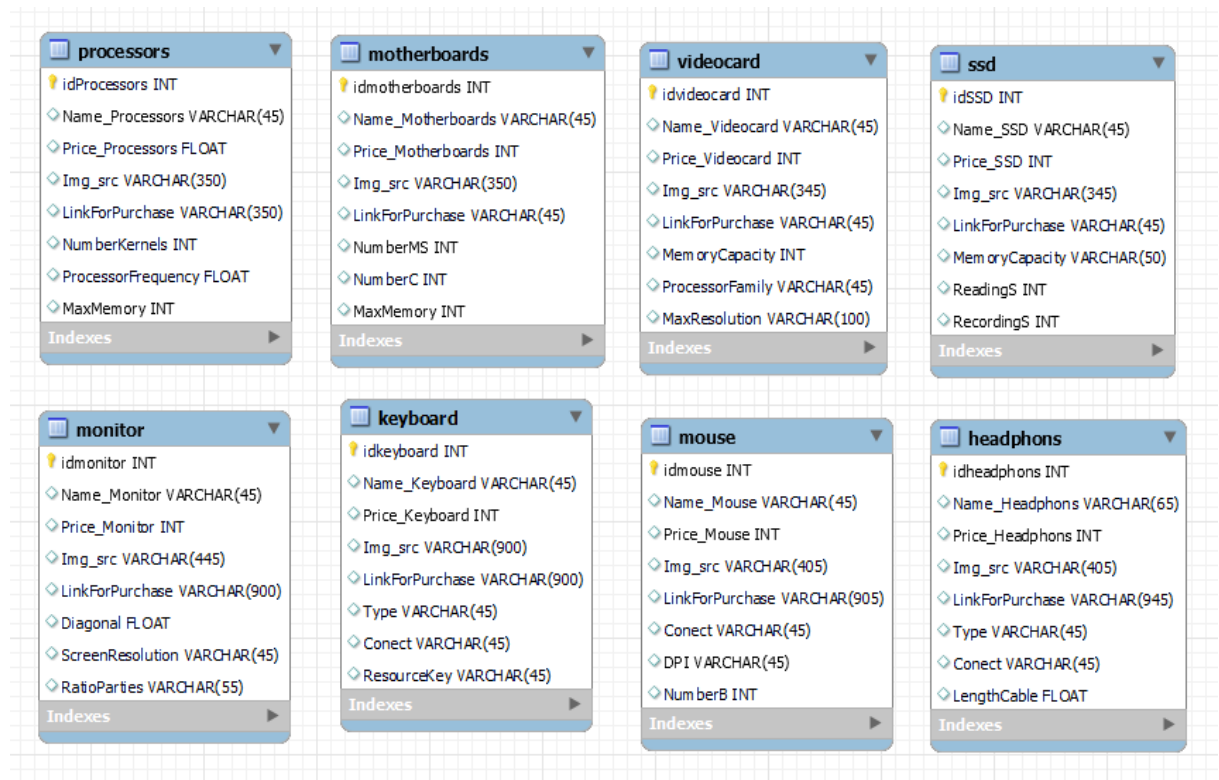


Рисунок 2.1 – Загальна структура бази даних

2.3 Розробка Wireframe

Для кращого розуміння функціональності розробленого проекту та візуального представлення всіх необхідних елементів було створено його Wireframe.(рис. 2.2).

Wireframe являє собою чорно-білий прототип застосунку, загалом викори-

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		31

стовується для кращого розуміння зовнішнього вигляду застосунку та розташування необхідних елементів з можливістю демонстрації переходів та примітивних анімацій.

Розроблений Wireframe, показує блоки елементів конфігурації, загальна структура сторінки, та можливий вигляд обрахунку ціни конфігурації та окремих елементів.

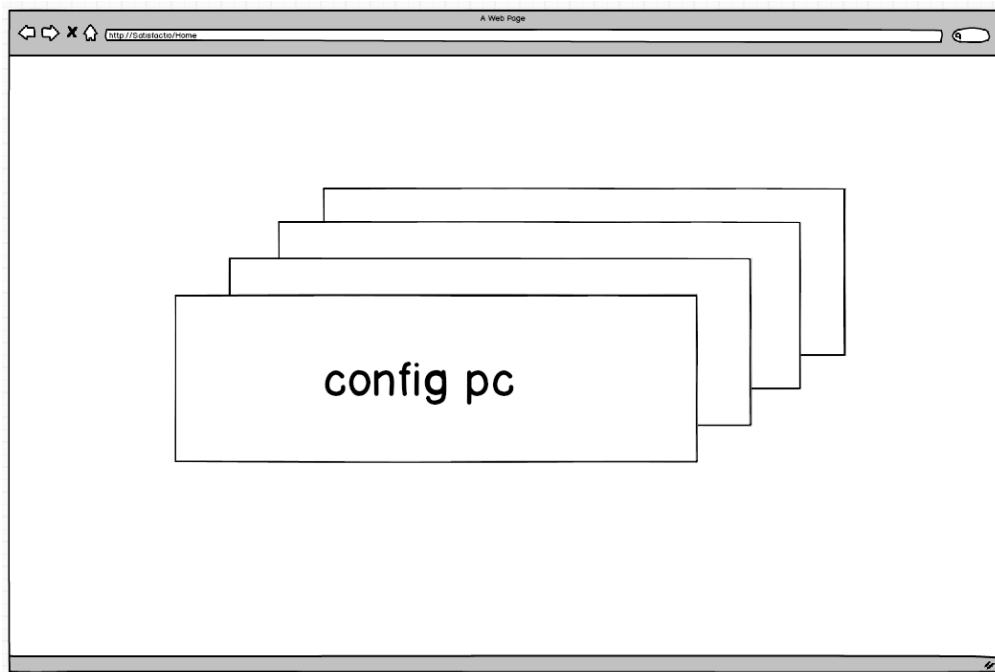


Рисунок 2.2 – Wireframe головної сторінки

Після створення Wireframe головної сторінки, точніше частини головної сторінки, відразу захотілось підкреслити стан блоків по центру, було прийнято рішення робити їх позицію фіксованою відносно заднього фону, що в житті створює ніби паралакс ефект.

Для подальшого проектування необхідно було вирішити як саме елементи будуть розташовуватись на сторінці, як буде виглядати підпис їх характеристик, ціна, посилання на магазини, як буде розгортатись список всіх елементів конкретної категорії конфігурації.

Відповідь отримаємо, переглянувши Wireframe розташування елементів (рис. 2.3).

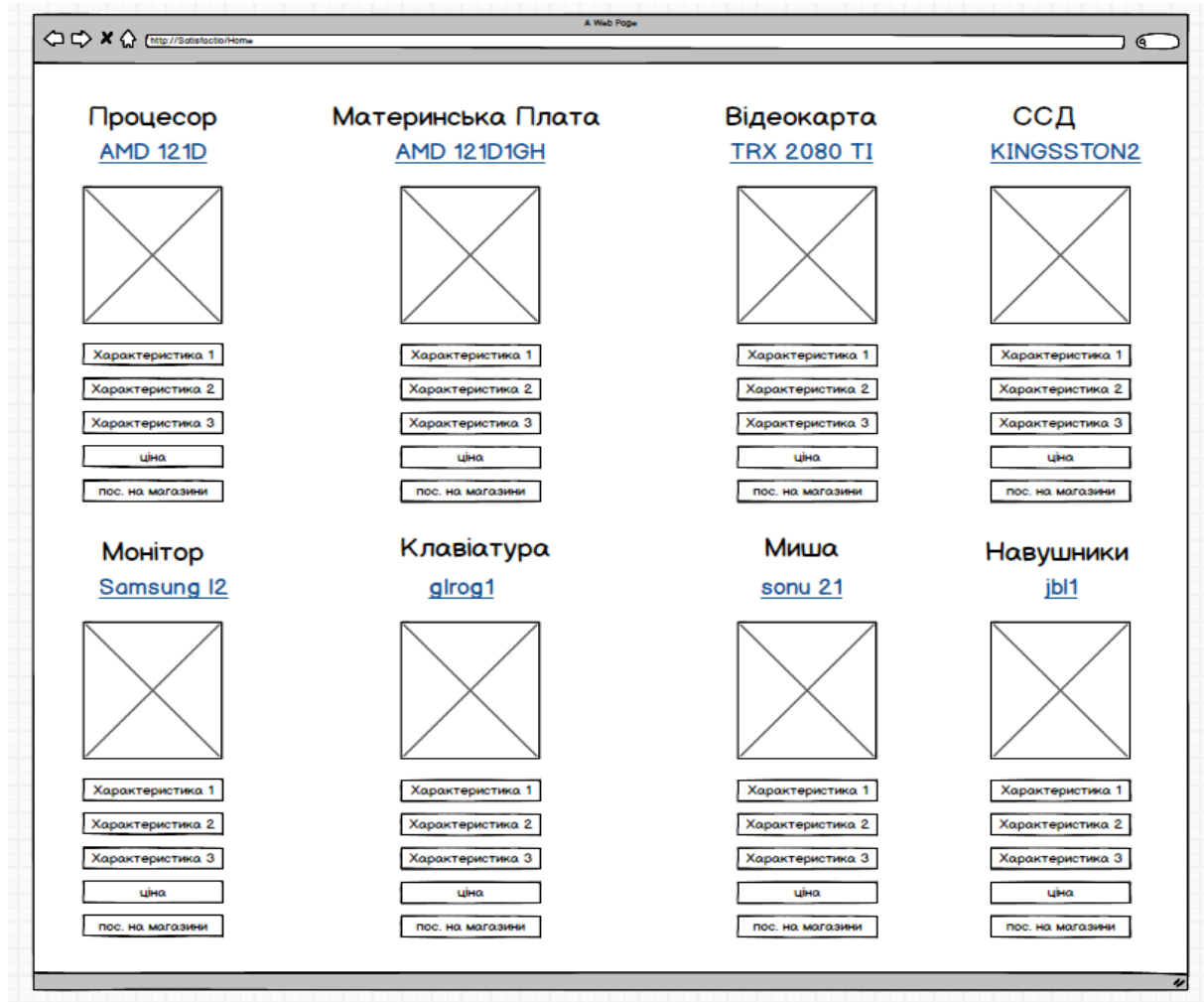


Рисунок 2.3 – Wireframe представлення елементів

Після отриманого результату стало очевидно, що цей варіант не підходить, необхідно було розробити концепцію враховуючи недоліки цієї спроби. Для початку необхідно розділити елементи на системні та периферію, для цього можна використати блок з текстом який не буде відходити від загального стилю але, буде підкреслювати тип елементів які розташовані після нього.

Наступним етапом буде збільшення шрифтів елементів та підбір фото з прозорим фоном, щоб виділити його сильніше на контрасті з заднім фоном сторінки.

Необхідно підкреслити список з елементом конфігурації, адже на Wireframe чітко видно що він зливається з характеристиками та стає непомітним. Посилання на магазин це останній елемент в блоці формули також необхідно або виділити спеціальним кольором або формою, щоб підкреслити що це посилання,

а не звичайний не інтерактивний блок. Враховуючи все вище описане, проблеми та методи їх вирішення було створено фінальний вар фрейм (рис.2.4).

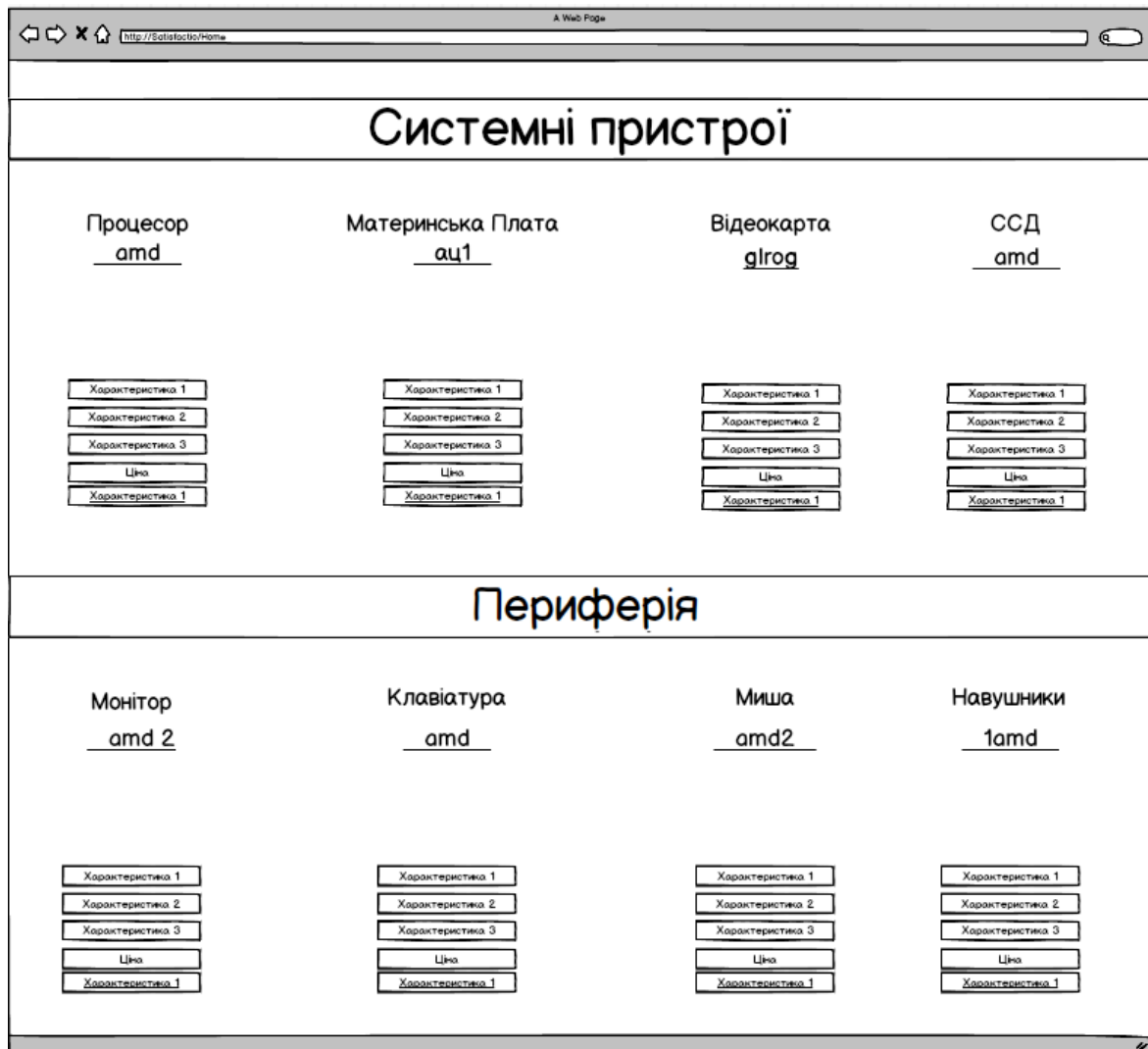


Рисунок 2.4 – Фінальний Wireframe відображення елементів конфігурації

Наступним для чого потрібно було виконати Wireframe це обрахування фінальної ціни та ціни кожного елементу конфігурації окремо. Як розділити ці дві ціни, та як візуально це представити.

Щоб було зрозуміло за що відповідає конкретний елемент, та яка його ціна(можливо, % від ціни) та загальна ціна її розташування та представлення. Потрібно розташувати кнопку для підрахунку всіх цих значень.

Для цього було спроектовано круговий графік (рис. 2.5), кожна частина якого буде відповідати за конкретний елемент конфігурації, при переміщенні

курсору на відповідну частину буде з'являтися назва конфігурації її тип, ціна та % від загальної ціни, що дозволить користувачу зрозуміти що в конфігурації найдорожче та найдешевше, та дасть змогу ефективніше підбирати збірку під свої фінансові можливості.



Рисунок 2.5 – Wireframe графіку

Висновки до розділу 2

В розділі було розглянуто ключові бібліотеки для роботи з базами даних в Java Script. Створено та налаштовано сервер в середовищі MySQL Workbench і створено базу даних.

В базі було створено необхідну кількість таблиць, налаштовано типи їх полів та проведено детальний опис запитів для створення таблиць мовою програмування SQL.

Оглянуто середовище для створення чорно-білих прототипів Balsamiq. Створено Wireframe проекту, описано та продемонстровано основні елементи застосунку.

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		35

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

3.1 Розробка сервера та запитів до бази

Для програмної реалізації вебзастосунку спочатку необхідно створити сервер який буде брати дані з бази, а тому розглянемо наступний код (рис. 3.1).

```
const mysqlConnection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "pc_config",
  multipleStatements: true,
});
```

Рисунок 3.1 – Код для з'єднання з базою даних

Це і є безпосереднє з'єднання до бази, ми вказуємо назву бази, користувача, пароль, та назву серверу. Запити до бази проводяться через HTTP запит «get» та перевірку на помилки у функції.

Для запиту з таблиці даних було створено для кожної таблиці свою адресу, а саме:

- Таблиця «motherboards» та адреса «localhost3003/motherboards»;
- Таблиця «videocard» та адреса «localhost3003/videocard»;
- Таблиця «ssd» та адреса «localhost3003/ssd»;
- Таблиця «monitor» та адреса «localhost3003/monitor»;
- Таблиця «keyboard» та адреса «localhost3003/keyboard»;
- Таблиця «mouse» та адреса «localhost3003/mouse»;
- Таблиця «headphons» та адреса «localhost3003/headphons».

Для звернення до бази даних, а конкретніше до кожної з таблиць було написано наступний код (рис. 3.2), який створює запит, отримує дані з бази та повертає їх на відповідну раніше вказану адресу.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		36

```

app.get("/processors", (req, res) => {
  mysqlConnection.query("SELECT * FROM processors", (err, results, fields) => {
    if (!err) res.send(results);
    else res.end(JSON.stringify(results));
  });
});
app.get("/motherboards", (req, res) => {
  mysqlConnection.query(
    "SELECT * FROM motherboards",
    (err, results, fields) => {
      if (!err) res.send(results);
      else res.end(JSON.stringify(results));
    });
});
app.get("/videocard", (req, res) => {
  mysqlConnection.query("SELECT * FROM videocard", (err, results, fields) => {
    if (!err) res.send(results);
    else res.end(JSON.stringify(results));
  });
});
app.get("/ssd", (req, res) => {
  mysqlConnection.query("SELECT * FROM ssd", (err, results, fields) => {
    if (!err) res.send(results);
    else res.end(JSON.stringify(results));
  });
});
});

```

Рисунок 3.2 – Код для запиту до перших 4 елементів конфігурації

Для графіка відношення цін нам потрібні назви всіх елементів та ціна з кожної таблиці. Цей запит (рис. 3.3) поверне нам необхідні дані у вигляді масиву даних за адресою «localhost3003/allname», в якому будуть імена та ціни всіх елементів з бази.

```

app.get("/allname", (req, res) => {
  mysqlConnection.query(
    "SELECT Name_Processors, Price_Processors,\
    Name_Motherboards, Price_Motherboards,\
    Name_Videocard, Price_Videocard,\
    Name_SSD, Price_SSD, \
    Name_Monitor, Price_Monitor,\
    Name_Keyboard, Price_Keyboard,\
    Name_Mouse, Price_Mouse, \
    Name_Headphons, Price_Headphons\
    FROM `processors`, `motherboards`, `videocard`, `ssd`, `monitor`, `keyboard`, `mouse`, `headphons` \
    WHERE idProcessors = idmotherboards \
    AND idProcessors = idvideocard \
    AND idProcessors = idSSD\
    AND idProcessors = idmonitor\
    AND idProcessors = idkeyboard\
    AND idProcessors = idmouse\
    AND idProcessors = idheadphons",
    (err, results, fields) => {
      if (!err) res.send(results);
      else res.end(JSON.stringify(results));
    }
  );
});

```

Рисунок 3.3 – Код для запиту

3.2 Стилiзацiя компонентiв за допомогою CSS

Розглянемо наступний CSS код (рис. 3.4) для хедеру, або першої сторiнки яку побачить користувач.

```
.header {  
  width: auto;  
  height: 1000px;  
  background: url(../img/backg.png) no-repeat;  
  background-size: auto;  
  background-color: #111111;  
  background-attachment: fixed;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-size: cover;  
}
```

Рисунок 3.4 – CSS для хедеру

Цей код, замiнює заднiй фон сторiнки на зображення та атрибути «attachment, position, repeat, cover» в сумi дають ефект схожий на ефект паралаксу, при перемiщеннi основнi блоки залишають свою позицiю змiщується тiльки заднiй фон, що виглядає доволi креативно та незвично (рис. 3.5).

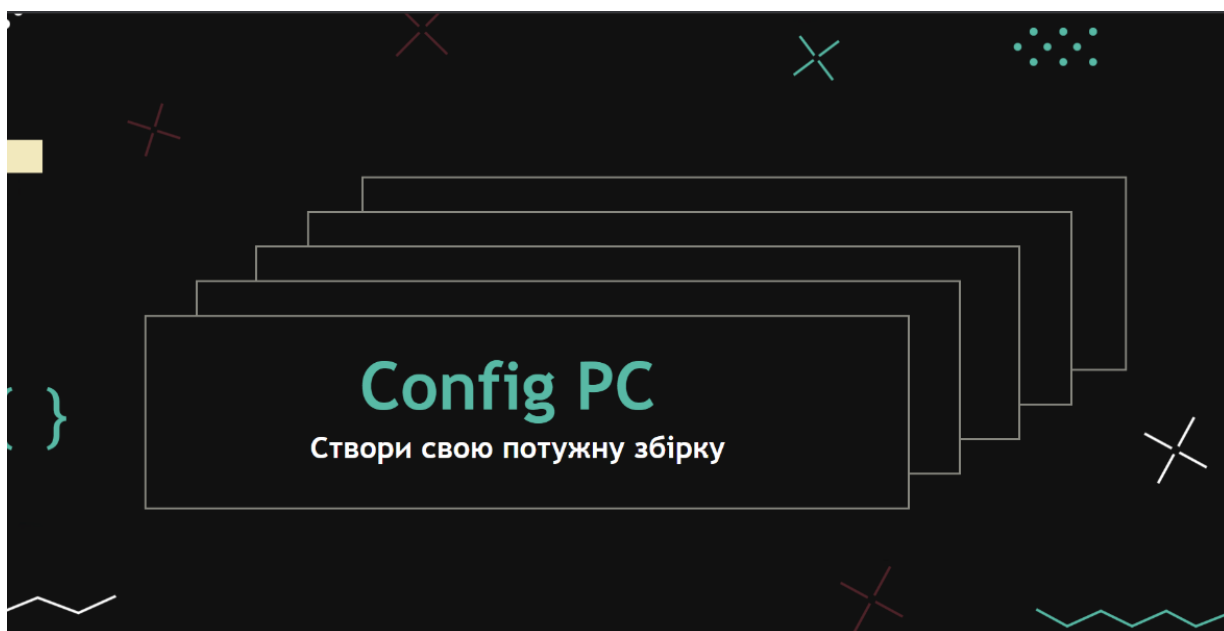


Рисунок 3.5 – Перша стилiзована сторiнка застосунку

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Пiдп.	Дата		38

Наступними необхідно створити дизайн для розділення елементів на системні пристрої та периферію. Підібравши кольори та необхідні розміри ми отримали наступний код (рис. 3.6).

```
.Body_element_Header№1,.Body_element_Header№2{
background-color: #111111;
border-bottom: #8884d8 2px solid;
border-top: #8884d8 2px solid;
width: auto;
height: 120px;
font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",
"Lucida Sans", Arial, sans-serif;
font-size: 100px;
text-align: center;
color: #58b9a5;
position: relative;
}
```

Рисунок 3.6 – Код стилізованих елементів розділення

Результат застосування коду цих стилів на (рис. 3.7).

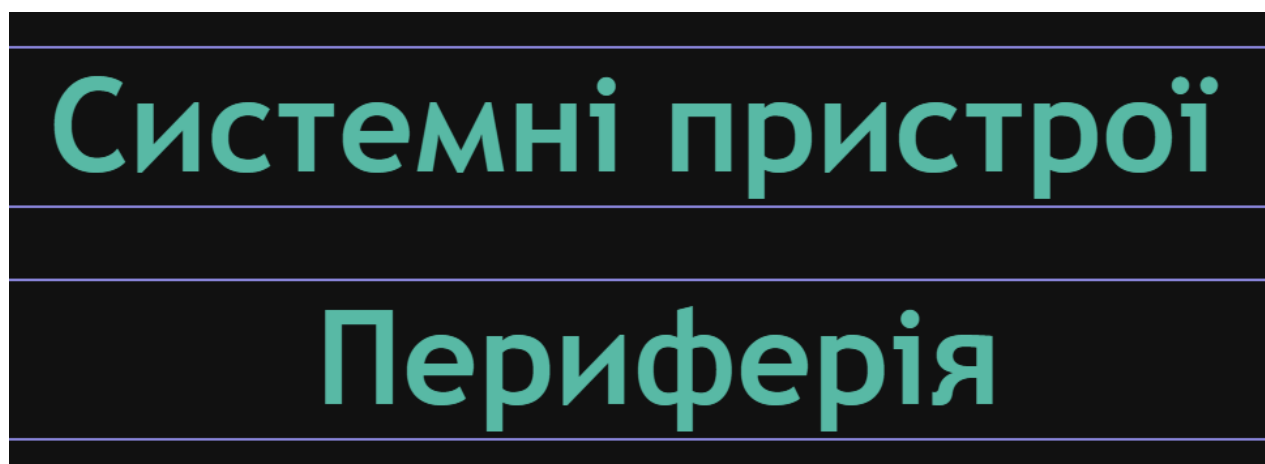


Рисунок 3.7 – Стилізовані елементи розділення

Далі нам необхідно стилізувати список, характеристики та фото елементу конфігурації.

Для фото було прийнято рішення шукати тільки ті в яких фон прозорий, для кращого візуального сприйняття фото.

Для списку було обрано колір, шрифт та стиль, наступний код (рис. 3.8).

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		39


```
.Processor_Select select {
  position: relative;
  text-align: left;
  width: 300px;
  top: -40px;
  font-size: 18px;
  border: 1px solid #111111;
  border-radius: 5px 5px 5px 5px;
  background-color: #111111;
  color: #8884d8;
  margin: 10% 10%;
  border-bottom: 1px solid #8884d8;
}
```

Рисунок 3.8 – Код для стилю списку

Для тексту елементів конфігурації було підбірано колір(рис. 3.9) який добре видно на чорному фоні та привертає до себе увагу, для списку було підбірано заголовки для процесорів «Intel» та «AMD» за допомогою атрибута «disabled» тегу «option».

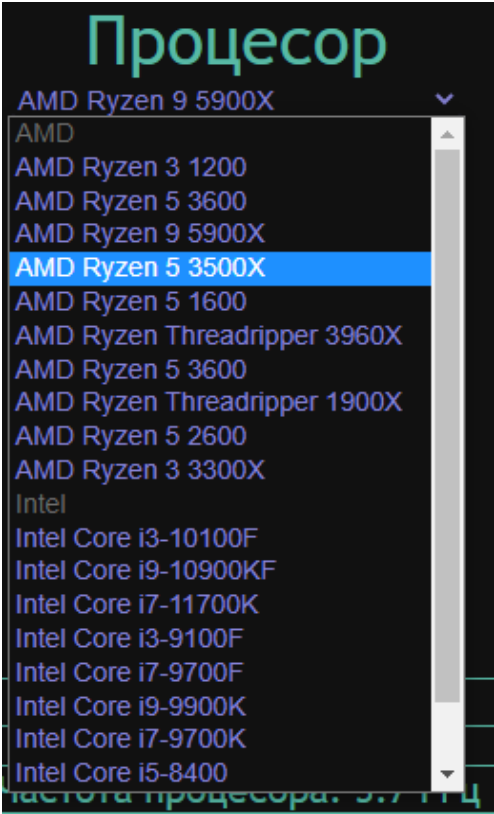


Рисунок 3.9 – Стилізований список елементів

Для зображення та характеристик було підібрано наступні стилі (рис. 3.10).

```
.Processor_Img {
  border-radius: 5px 5px 5px 5px;
  position: relative;
  top: -85px;
  width: 575px;
  height: 350px;
  overflow: hidden;
}
.Technical_characteristics {
  font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",
  "Lucida Sans", Arial, sans-serif;
  position: relative;
  top: -60px;
  font-size: 25px;
  width: 400px;
  list-style-type: none;
  color: #58b9a5;
}
```

Рисунок 3.10 – Код для стилю характеристик та зображення

Для зображення та характеристик було підібрано колір який також добре видно на чорному фоні та не конфліктує з кольором списку елементів. Для посилення на магазини було створено підкреслення, щоб якось виділити цей елемент з блоку (рис. 3.11)

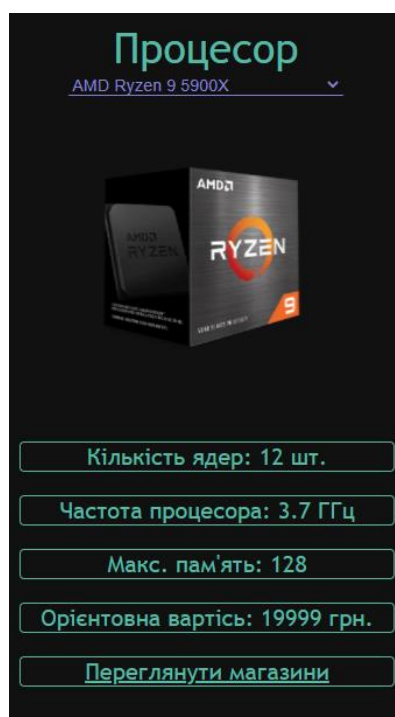


Рисунок 3.11 – Стилізований блок елемента

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		41

3.3 Програмна реалізація компонентів для конфігурації

Було створено створили сервер, тепер необхідно вибрати дані та помістити їх у список кожного з елементів, ціну посилання на магазин та характеристики також необхідно показати в певних полях сторінки.

Розглянемо частину коду (рис. 3.12) для першого елемента сторінки, процесора.

```
import React, { useState, useEffect, useRef } from "react";
import "./processors.css";
export let Processor_Index;
export default function Processor() {
  const [data, setData] = useState(null);

  const d = useRef();
  const [img, setimg] = useState("");

  useEffect(() => {
    Processor_Index = d.current?.value;
  }, [d.current?.value]);

  useEffect(() => {
    fetch("http://localhost:3003/processors")
      .then((payload) => {
        const a = payload.json();
        return a;
      })
      .then((a) => {
        setData(a);
        console.log(a);
      });
  }, []);
  if (!data) {
    return (
      <>
      <div> Loading.</div>
      </>
    );
  }
}
```

Рисунок 3.12 – Код першого елемента сторінки

Тут і в інших компонентах ми використовуємо різноманітні хуки бібліотеки React, Use State це хук яки допомагає динамічно змінити значення змінної через функцію SetState.

Use Ref використовуємо для зберігання об'єкту, в нашому випадку це

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		42

атрибут «Value», точніше його індекс, тега «Select» який необхідний нам для коректного зображення того чи іншого елемента списку.

«Use Effect» потрібен нам для змін коли якийсь об'єкт змінюється, у вище написаному коді «Use Effect» буде спрацьовувати кожен раз коли буде мінятися змінна «Use Ref», тобто «d» він буде змінній «Processor_Index» присвоювати значення «d», ця змінна нам потрібна для експорту та відображення результату обчислення ціни, метод якого буде розглядатись далі по виконанню роботи.

Розглянемо код для реалізації візуалізації ціни, магазинів, та характеристик процесора (рис. 3.13).

```
<ul className="Technical_characteristics">
  <li>Кількість ядер: {data[d.current?.value]?.NumberKernels} шт.</li>
  <li>
    | Частота процесора: {data[d.current?.value]?.ProcessorFrequency} ГГц{" "}
  </li>
  <li>Макс. пам'ять: {data[d.current?.value]?.MaxMemory}</li>
  <li>
    | Орієнтовна вартість: {data[d.current?.value]?.Price_Processors} грн.
  </li>
  <li>
    <a href={data[d.current?.value]?.LinkForPurchase} target="blank">
      | Переглянути магазини
    </a>
  </li>
</ul>
```

Рисунок 3.13 – Код для ціни, магазинів та характеристик

Ми використовуємо тег «ul» що відповідає за створення списку, «li» це рядок списку.

Наступний код «{data[d.current?.value]?.NumberKernels}» відповідає за вибір даних з масиву «data» який ми створили виконавши метод «fetch».

Якщо повністю розшифрувати код, то він працює наступним чином, ми з масиву «data» обираємо елемент під індексом «d» який ми отримали за допомогою «Use Ref» з списку, тобто «d.current?.value» = «value="n"», після чого ми знаком «?» перевіряємо чи існують дані, якщо так, то Javs Script виведе їх (рис. 3.14).

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		43



Рисунок 3.14 – Результат вибору елемента конфігурації

3.4 Програмна реалізація виводу графіку ціни

Для розбиття загальної ціни на ціну кожного елемента конфігурації та вивід цього за допомогою графіка використаємо бібліотеку для графіків «recharts».

Розглянемо наступний код (рис. 3.15):

```
import React from "react";
import { useEffect, useState } from "react";
import "./result.css";
import { Processor_Index } from "./Processors";
import { Motherboard_Index } from "./Motherboards";
import { Videocard_Index } from "./VideoCard";
import { SSD_Index } from "./SSD";
import { Monitor_Index } from "./Monitor";
import { Keyboard_Index } from "./Keyboard";
import { Mouse_Index } from "./Mouse";
import { Headphone_Index } from "./Headphone";
import { PieChart, Pie, Sector, ResponsiveContainer } from "recharts";
```

Рисунок 3.15 – Імпортовані дані

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		44

Загалом тут імпортовані дані з інших компонентів програми для отримання їх індексу. Імпорт «PieChart» з бібліотеки «recharts» це і є кругова діаграма, наступний код це створення хуків «Use State» для кожної з імпортованих змінних (рис. 3.16), щоб мати змогу динамічно їх змінювати.

```
const [state, setState] = useState(0);
const [data, setData] = useState(null);
const [Processor, setProcessos] = useState(0);
const [Motherboard, setMotherboard] = useState(0);
const [Videocard, setVideocard] = useState(0);
const [SSD, setSSD] = useState(0);
const [Monitor, setMonitor] = useState(0);
const [Keyboard, setKeyboard] = useState(0);
const [Mouse, setMouse] = useState();
const [Headphone, setHeadphone] = useState();
```

Рисунок 3.16 – Функції для динамічної зміни значень

Далі необхідно створити масив (рис. 3.17) з даними бази, для цього використаємо метод «fetch» та хук «Use Effect» щоб метод працював тільки при умові що дані змінились:

```
useEffect(() => {
  fetch("http://localhost:3003/allname")
    .then((payload) => {
      const a = payload.json();
      return a;
    })
    .then((a) => {
      setData(a);
    });
}, []);
if (!data) {
  return (
    <>
    <div> Loading..</div>
    </>
  );
}
```

Рисунок 3.17 – Створення масиву даних

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		45

Далі необхідно кнопки «Оновити» яка відповідає за оновлення даних для графіку, присвоїти функцію (рис. 3.18)

```
const evtAlo = () => {
  setProcessos(Processor_Index);
  setMotherboard(Motherboard_Index);
  setVideocard(Videocard_Index);
  setSSD(SSD_Index);
  setMonitor(Monitor_Index);
  setKeyboard(Keyboard_Index);
  setMouse(Mouse_Index);
  setHeadphone(Headphone_Index);
  return;
};
```

Рисунок 3.18 – Присвоєння події кнопки

Ця функція по натиску на кнопку «оновити» поновлює значення наших змінних, передаючи їм значення імпортованих змінних через функцію хука SetState. Наступний крок це створення масиву даних для графіку (рис. 3.19), та змінну яка буде зберігати загальну ціну конфігурації.

```
const data1 = [
  { type: "Процесор", name: data[Processor]?.Name_Processors ? data[Processor]?.Name_Processors : "Не обрано", value: data[Processor]?.Price_Processors ? data[Processor]?.Price_Processors : 0 },
  { type: "Материнська Плата", name: data[Motherboard]?.Name_Motherboards ? data[Motherboard]?.Name_Motherboards : "Не обрано", value: data[Motherboard]?.Price_Motherboards ? data[Motherboard]?.Price_Motherboards : 0 },
  { type: "Відеокарта", name: data[Videocard]?.Name_Videocard ? data[Videocard]?.Name_Videocard : "Не обрано", value: data[Videocard]?.Price_Videocard ? data[Videocard]?.Price_Videocard : 0 },
  { type: "СSD", name: data[SSD]?.Name_SSD ? data[SSD]?.Name_SSD : "Не обрано", value: data[SSD]?.Price_SSD ? data[SSD]?.Price_SSD : 0 },
  { type: "Монітор", name: data[Monitor]?.Name_Monitor ? data[Monitor]?.Name_Monitor : "Не обрано", value: data[Monitor]?.Price_Monitor ? data[Monitor]?.Price_Monitor : 0 },
  { type: "Клавіатура", name: data[Keyboard]?.Name_Keyboard ? data[Keyboard]?.Name_Keyboard : "Не обрано", value: data[Keyboard]?.Price_Keyboard ? data[Keyboard]?.Price_Keyboard : 0 },
  { type: "Миша", name: data[Mouse]?.Name_Mouse ? data[Mouse]?.Name_Mouse : "Не обрано", value: data[Mouse]?.Price_Mouse ? data[Mouse]?.Price_Mouse : 0 },
  { type: "Наушники", name: data[Headphone]?.Name_Headphons ? data[Headphone]?.Name_Headphons : "Не обрано", value: data[Headphone]?.Price_Headphons ? data[Headphone]?.Price_Headphons : 0 }
];
```

Рисунок 3.19 – Дані для графіку

В масиві «data1» зберігається три основні значення, «type» відповідає за підпис конкретного елемента конфігурації, «name» за назву елемента, «value» за ціну.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підп.	Дата		

Проводиться перевірка чи є дані в масиві знаком «?» та присвоюємо їх значення змінним масиву.

Змінна «TotalPrice» (рис. 3.20) отримує значення з графіка та виводить загальну ціну.

```
const TotalPrice = data1[0].value +
  data1[1].value + data1[2].value +
  data1[3].value + data1[4].value +
  data1[5].value + data1[6].value +
  data1[7].value
```

Рисунок 3.20 – Код загальної ціни

Нище в коді ми задаємо основні значення графіку колір, радіус, основні підписи (рис. 3.21).

```
<g>
  <text x={cx} y={cy} dy={8} textAnchor="middle" fill={fill} fontSize="50px">
    {payload.value}
  </text>
  <Sector
    cx={cx}
    cy={cy}
    innerRadius={innerRadius}
    outerRadius={outerRadius}
    startAngle={startAngle}
    endAngle={endAngle}
    fill={fill}
  />
  <Sector
    cx={cx}
    cy={cy}
    startAngle={startAngle}
    endAngle={endAngle}
    innerRadius={outerRadius + 6}
    outerRadius={outerRadius + 10}
    fill={fill}
  />
  <path d={`M${sx},${sy}L${mx},${my}L${ex},${ey}`} stroke={fill} fill="none" />
  <circle cx={ex} cy={ey} r={2} fill={fill} stroke="none" />
  <text x={ex + (cos >= 0 ? 1 : -1) * 12} y={ey} textAnchor={textAnchor} fill="#58b9a5" fontSize="30px">{` ${name}`}</text>
  <text x={ex + (cos >= 0 ? 1 : -1) * 12} y={ey} dy={18} textAnchor={textAnchor} fontSize="20px" fill="#999" >
    {` ${type}: ${((percent * 100).toFixed(2))% ціни`}
  </text>
</g>
```

Рисунок 3.21 – Код графіку

Через те, що частина коду графіка (рис. 3.22) є конструкцією бібліотеки

					КР.ІІЗ – 22.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		47

«recharts» сильних змін в ньому не робилось, щоб не порушувати його конструкцію та функціональність.

Змінювались тільки дані які відповідають за візуальний вигляд і поділ графіку на секції.

Було додано ще декілька підписів та вивід загальної ціни в середині блоку графіка, але загалом код залишився в «чистому вигляді».

```
return (  
  <>  
  <div className="ResultConfig">  
    <div className="bgresult">  
      <h1 className="result_header"> Загальна ціна: {TotalPrice ? TotalPrice : 0 } грн. </h1>  
      <button onClick={evntAlo} className="result_Button">  
        { " " }  
        Оновити{ " " }  
      </button>  
      <ResponsiveContainer width="100%" height="100%" aspect={-1}>  
        <PieChart width={500} height={500}>  
          <Pie  
            activeIndex={state.activeIndex}  
            activeShape={renderActiveShape}  
            data={data1}  
            cx="50%"  
            cy="35%"  
            innerRadius={140}  
            outerRadius={160}  
            fill="#8884d8"  
            dataKey="value"  
            onMouseEnter={onPieEnter}  
          />  
        </PieChart>  
      </ResponsiveContainer>  
      <div className="linefordesign"></div>  
    </div>  
  </>  
)  
);
```

Рисунок 3.22 – Головний код графіку

Загалом проектування графіку допомогло зрозуміти деякі не потрібні речі в додатку.

Під час його проектування та створення було оптимізовано код, та складено умовний концепт майбутнього вдосконалення функціоналу як графіку, так і інших компонентів системи використовуючи вже готові конструкції, що не тільки полегшує роботу, а ще й покращує результат.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		48

Загальний вигляд графіку та приклад зображення результату на (рис. 3.23)

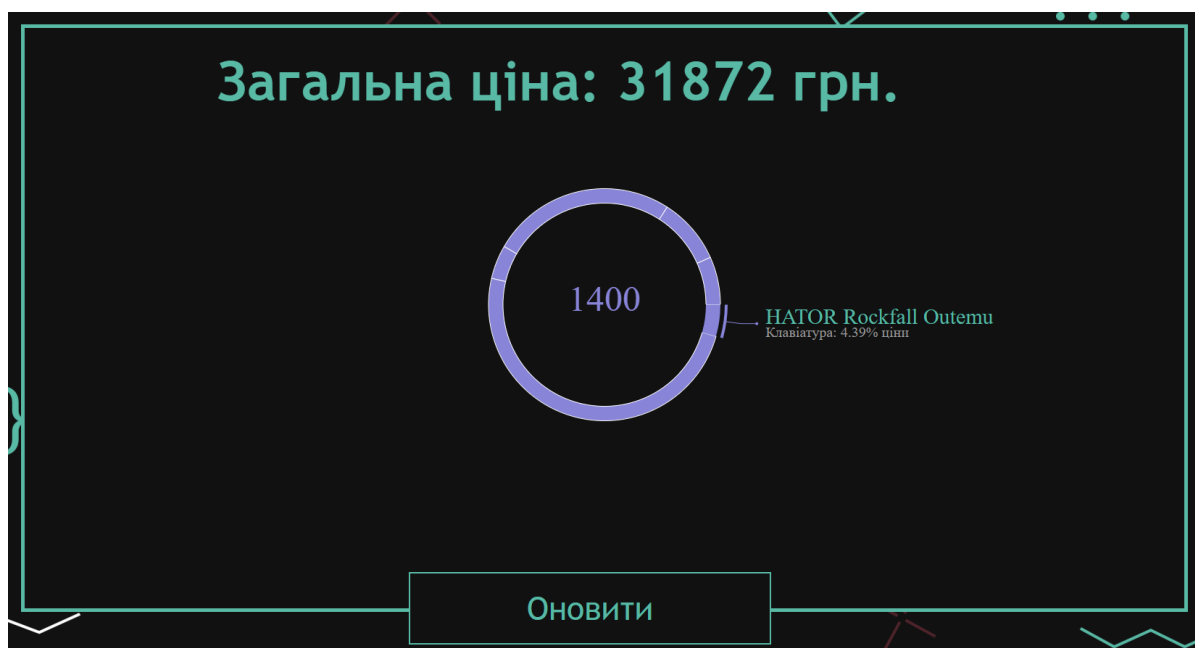


Рисунок 3.23 – Зображення графіка

Висновки до розділу 3

В розділі було детально розглянуто та описано створення сервера в Java Script та проаналізовано ключові запити до сервера та бази, отримання та обробку відповіді.

Розроблено дизайн застосунку та його ключових складових за допомогою CSS та продемонстровано результати стилізації.

Розроблено та розглянуто код основних елементів застосунку, розглянуто вибір елемента, передачу даних між списком і блоком.

Продемонстровано стилізацію фото блоку та характеристики.

Було проведено пояснення хуків які були використані в застосунку та їх особливостей у використанні.

Створено дизайн та код графіку загальної ціни компонентів конфігурації комп'ютера, проаналізовано код та ключові складові бібліотеки «Recharts», за допомогою якої і було створено та налаштовано графік.

Проведено демонстрацію фінального результату.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		49

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Правила безпечної роботи з комп'ютером

Перш за все у нас повинні бути правильний стіл і стілець з регульованими по висоті ніжками. Бажано, щоб спинка стільця була вигнута вперед, щоб упиратися на неї спиною, а не сидіти, розвалившись у кріслі.

Сидіти потрібно прямо, так, щоб між стегном і спиною був прямий кут. Не потрібно схрещувати ноги під стільцем, ступня повинна упиратися в підлогу і стояти на підставці.

Лікті слід покласти на стіл в горизонтальному положенні, тобто між ліктем і передпліччям повинен бути знову-таки прямий кут. І приберіть ліву руку від обличчя! Друге правило, монітор потрібно розташовувати так, щоб діставати витягнутою рукою екран і дивитися в нього прямо. Рівень очей повинен перебувати на рівні верхньої частини або трохи нижче, але ні в якому разі не вище, щоб не було додаткового навантаження на очі.

Раніше монітор стояв у мене на спеціальній підставці, зараз я відмовилася від неї і коли сідаю за чужий комп'ютер з такою підставкою, відразу відчуваю дискомфорт. А раніше не помічала, що сиділа неправильно. Ще не менш важливо освітлення. Монітор як настільна лампа, з тією ж інтенсивністю світить нам в очі. І вже через годину роботи за комп'ютером очі втомлюються, можуть почервоніти повіки, з'явитися різь і печіння в очах.

Монітор потрібно ставити так, щоб на нього не падав денне світло. А освітленість кімнати повинна збігатися з освітленістю екрана. Вікна краще закривати за допомогою жалюзі. Слід уникати білого забарвлення стін, яка додатково навантажує очі.

Правило третє якою повинна бути клавіатура? Клавіатура не повинна розташовуватися дуже далеко від краю столу. Через це руки можуть відчувати надмірну напругу, і їх судомить.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		50

Багато використовувати ергономічну клавіатуру зігнутої форми, з поділом під кожен руку і з підставкою під зап'ястя, щоб розвантажити кисті. Руки тримайте паралельно один одному, стежте, щоб лікті розташовувалися під прямим кутом до передпліччя.

Правило четверте комп'ютерна миша, використовуйте ергономічну бездротову мишу зручної форми за розміром своєї долоні, таку, щоб кисть зберігала своє природне положення.

Обов'язково використовуйте килимок для миші, краще з гелевою подушкою, яка знижує тиск на зап'ясті. Кисть руки при роботі з мишею повинна бути розташована паралельно кришці столу.

А взагалі краще якомога менше користуйтеся мишею, використовуйте гарячі клавіші клавіатури. Особливо при роботі в текстових редакторах я часто спостерігаю, що багато хто користується мишею, коли набагато зручніше використовувати клавіші для переміщення по тексті і інших команд редактора.

Основні правила роботи за комп'ютером безпосередньо сидячи за комп'ютером, ми розглянули. Але є ще одне дуже важливе правило.

П'яте правило роботи за комп'ютером, мабуть, найголовніше за інших. Якщо попередні з якихось причин не завжди вдається виконувати, то відпочинок потрібно влаштовувати обов'язково, щоб зняти навантаження і зі спини, і з зап'ясть, і з очей.

Через кожен годину роботи влаштовуйте 15-ти хвилинний відпочинок. У цей час можна і чайку попийти (але не за комп'ютером!), А ще краще зробити кілька фізичних вправ, в тому числі для шиї, рук і очей.

Стисніть кисті в кулак, потримайте 10 секунд і розтисніть. Зігніть пальці назад, потримайте 10 секунд, розслабте.

Подивіться вдаль, краще на якийсь предмет зеленого кольору, наприклад, на зелень за вікном, потім переведіть погляд на кінчик носа. Повторіть кілька разів.

Закрийте повіки і зробіть кілька обертальних рухів зіницями за годинниковою стрілкою і проти.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		51

4.2 Рекомендовані умови для роботи за комп'ютером

Для забезпечення комфортної та що головніше не шкідливої роботи за комп'ютером слід притримуватись наступних умов:

1. Сидіть глибоко на твердому стільці з високою спинкою, що має вигин для попереку, це вирівняє спину і дасть підтримку шиї. Край стільця не повинен тиснути на судини під колінами.
2. Відстань до монітора повинна бути 50-70см.
3. Використовуйте мишку відповідних розмірів, зручної форми.
4. Робіть перерву в сидячій роботі, вставайте і ходіть 15-20 хвилин кожні 1-2 години.
5. Правильно організуйте освітлення робочого місця. При слабкому світлі очі напружуються і болять. Стримайте яскравість екрану. Літери і цифри на екрані це маленькі світлові промені, які йдуть прямо в очі. Потрібно відрегулювати їх контрастність, щоб світло не був дуже яскравим.
6. Закривайте очі для відпочинку. Час від часу відводите очі вбік, щоб дати відпочити своєму зору.
7. Переміщайте погляд по всій площі екрану, намагайтеся не дивитися напружено в одну точку. Нехай поперемінно працюють всі м'язи очей, а не окремі групи, на які в цьому випадку буде падати максимальне навантаження.
8. Використовуйте спеціальний очний гель (Визин, Видисик), який запобігає «обсушування» рогівки ока. Запобігти захворювання можуть окуляри для роботи за комп'ютером, які має особливе покриття.

4.3 Розрахунок захисного заземлення

Для запобігання чи створенню значної напруженості та захисту від статичної електрики у приміщеннях з комп'ютерною технікою повинні бути заземлені (занулені) всі електричні прилади.

Підлога повинна мати антистатичне покриття. Захист від статичної елект-

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		52

рики повинно проводитись згідно з санітарно-гігієнічними нормами.

Під захисним заземленням розуміють з'єднання металевих струмоведучих частин електроустановок із землею через заземлюючі провідники та заземлювачі для створення між цими частинами і землею малого опору. Для захисного заземлення застосовують заземлювачі із труб. Це декілька металевих труб діаметром 25-50 мм, довжиною 2-3 м, які забивають в землю на відстані 4-6 м одна від одної і з'єднують між собою металічною трубою. Остання проходить в приміщенні, з'єднується з внутрішнім контуром. Розрахунок зводиться до визначення необхідної кількості труб заземлювача і довжини металічної труби, яка з'єднує вертикальні заземлювачі.

4.4 Пожежна безпека

Комп'ютерні класи призначені для проведення в них роботи та занять студентів, пов'язаних з використанням персональних комп'ютерів, які являються об'єктами підвищеної пожежної небезпеки.

Площа приміщення на одне робоче місце повинна становити 6 кв. м, а об'єм не менше ніж 20 куб. м. До безпосередньої роботи з ПК повинні допускатися особи, що не мають медичних протипоказань.

Приміщення повинні обладнуватися системою кондиціонування повітря чи ефективною приточно-витяжною вентиляцією.

Приміщення для роботи з ПК повинні бути забезпечені аптечкою для надання першої медичної допомоги та первинними засобами пожежогасіння.

Первинні засоби пожежогасіння це:

- Переносні вогнегасники;
- Пожежний інвентар;
- Піщані матеріали;
- Аерозольні генераторні вогнегасники;
- Пожежні крани;
- Спеціальні покривала.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		53

Найкраще в комп'ютерних класах та лабораторіях застосовувати порошкові вогнегасники типів ВП-1, ВП-2, ВП-5, як найбільш безпечні та ефективні.

Кожне приміщення повинне бути обладнане мінімум одним вогнегасником, об'єм якого залежить від площі приміщення.

Висновки до розділу 4

В розділі були описані всі необхідні умови для того щоб зробити експлуатацію ПК комфортною та не шкідливою. Було описано правила поведінки за комп'ютером щоб уникнути пожежі, рекомендації що до умов робочого місця та правила безпечної роботи з комп'ютером.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		54

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено застосунок для підбору компонентів ПК. Створено та стилізовано сучасний дизайн. Розроблено блок елемента в якому розташовані: характеристики, ціна, назва, фото, посилання на магазин. Характеристики, назва та рекомендована ціна беруться з бази даних для якої було спроектовано сервер для роботи з даними.

Розроблено графік який адаптивно реагує на зміну ціни кожного елемента в конфігурації, та виводить загальну ціну. Було зібрано посилання на магазини до кожного елемента конфігурації в базі.

Було проведено тестування на роботоспроможність проєкту, яку він пройшов на відмінно.

Проєкт залишає за собою можливість вдосконалення та оновлення функціоналу.

Розглянуто заходи з охорони праці при роботі з ПК.

					КР.ІІЗ – 22.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		55

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Радянський винахідник Горохів придумав комп'ютер ще до Стіва Джобса digger.ru: веб-сайт. URL: digger.ru/news/sovetskij
2. Конфігуратор BMW bmw.ua: веб-сайт. URL: configure.bmw.ru
3. Персональний комп'ютер *Вікіпедія*: веб-сайт. URL: uk.wikipedia.org/wiki/Персональний_комп'ютер
4. Перший комп'ютер Apple продали за 355 тисяч доларів *lenta.ua*: веб-сайт. URL: lenta.ua/news/2017/06/20/apple_i/#:~:text=Комп'ютер%20был%20разработан%20и%20собран,любил%20«повторяющиеся%20цифры»).
5. Рефакторинг *Вікіпедія*: веб-сайт. URL: uk.wikipedia.org/wiki/Рефакторинг
6. Visual Studio Code *Вікіпедія*: веб-сайт. URL: a.wikipedia.org/wiki/Visual_Studio_Code
7. Мокап, вайрфрейм, прототип і скетч: ху із ху *blog.sibirix.ru*: веб-сайт. URL: blog.sibirix.ru/2017/08/29/mockup-wireframe-prototype-sketch/
8. Balsamiq *Вікіпедія*: веб-сайт. URL: https://360wiki.ua/wiki/Balsamiq
9. Введення: знайомство з React *ua.reactjs.org*: веб-сайт. URL: ua.reactjs.org/tutorial/tutorial.html
10. MySQL *Вікіпедія*: веб-сайт. URL: ru.wikipedia.org/wiki/MySQL
11. React *Вікіпедія*: веб-сайт. URL: ua.wikipedia.org/wiki/React
12. SQL *Вікіпедія*: веб-сайт. URL: ua.wikipedia.org/wiki/SQL
13. База даних *Вікіпедія*: веб-сайт. URL: uk.wikipedia.org/wiki/База_даних
14. Робота з базою даних *medium.com*: веб-сайт. URL: medium.com/devschacht/node-hero-chapter-5-dd79607858f2
15. Кривенко, Г. М. Охорона праці : метод. вказівки до викон. розд. диплом. роботи бакалавра / Г. М. Кривенко, І. І. Стеліга.-Івано-Франківськ : ІФНТУНГ, 2015.-86 с.

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		56

16. В.І. Голінько, М.Ю. Іконніков, Я.Я. Лебедев «Охорона праці в галузі інформаційних технологій»

					КР.ПЗ – 22.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		57

