

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА  
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

**РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПІДБОРУ КОМП'ЮТЕРНИХ  
КОМПЛЕКТУЮЧИХ**

**Текстова частина до курсової роботи  
за спеціальністю „Комп'ютерні науки”**

Керівник курсової роботи  
ас. Калітовський Б.В.

\_\_\_\_\_  
(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

Виконав студент \_\_\_\_\_

Березніков Олексій

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

**Київ 2021**

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА  
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,

доцент, к.ф-м.н.

\_\_\_\_\_ Жежерун О.П.

(підпис)

„\_\_\_\_\_” \_\_\_\_\_ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту 3 року навчання БП «Комп'ютерні науки»

Березнікову Олексію Дмитровичу

ТЕМА: Розробка вебдодатку для підбору комп'ютерних комплектуючих  
Вихідні дані: Онлайн сервіс, розроблений на мові Typescript та Java, що  
взаємодіє з СКБД PostgreSQL через вебінтерфейс IntelliJ Idea.

Індивідуальне завдання

Вступ

1. Аналіз предметної області. Формулювання завдання курсової роботи
2. Теоретичні відомості
3. Опис реалізації програмного продукту

Висновки

Список використаних джерел

Додатки (за необхідністю)

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2021 р. Керівник \_\_\_\_\_ (підпис)

Завдання отримав \_\_\_\_\_ (підпис)

**Тема:** Розробка вебдодатку для підбору комп'ютерних комплектуючих

**Календарний план виконання роботи:**

№ п/п	Назва етапу дипломного проєкту (роботи)	Термін виконання етапу	Примітки
1.	Отримання теми курсової роботи	11.10.2020	
2.	Огляд літератури за темою роботи	15.01.2021	
3.	Аналіз схожих додатків	29.02.2021	
4.	Створення та розгортання бази даних	15.03.2021	
5.	Програмування розробленого алгоритму	16.03.2021	
6.	Тестування розробленого алгоритму	10.04.2021	
7.	Програмування вебінтерфейсу	20.04.2021	
8.	Корегування роботи з зауваженнями наукового керівника	02.05.2021	
9.	Написання текстової частини	07.05.2021	
10.	Створення презентації	10.05.2021	
11.	Захист курсової роботи	18-29.05.2021	

Студент \_\_\_\_\_ Березніков О.Д. \_\_\_\_\_

Керівник \_\_\_\_\_ Калітовський Б.В. \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

**ЗМІСТ**

<b>Анотація .....</b>	<b>5</b>
<b>ВСТУП .....</b>	<b>7</b>
<b>РОЗДІЛ 1: Аналіз предметної області.....</b>	<b>11</b>
1.1 Обґрунтування актуальності задачі .....	11
1.2 Огляд схожих додатків по підбору комп'ютерних комплектуючих.....	11
1.3 Опис вимог до розробки додатку.....	12
1.4 Висновок до розділу 1.....	13
<b>РОЗДІЛ 2: Теоретичні відомості.....</b>	<b>14</b>
2.1. Компоненти комп'ютера та їх сумісність .....	14
2.2 Типи комп'ютерних збірок .....	21
2.3 Переваги обраної СКБД та сервісу де розгорнутий додаток .....	21
2.4 Висновок до розділу 2.....	23
<b>РОЗДІЛ 3: Проектування та розробка додатку .....</b>	<b>24</b>
3.1 Опис технічного завдання на розробку .....	24
3.2 Обґрунтування обраних засобів розробки .....	27
3.3 Опис реалізації .....	32
3.4. Опис реляційної моделі .....	32
3.5 Розробка головного сервісу.....	42
3.6 Тестування програми і результати її виконання .....	44
3.7 Висновок до розділу 3.....	52
<b>Висновки .....</b>	<b>53</b>
<b>Список використаної літератури.....</b>	<b>54</b>

## **Анотація**

Роботу присвячено розробці вебдодатку, який перевіряє комп'ютерні комплектуючі на сумісність, пропонує зміни засновуючись на ефективності, лагодить несумісні збірки, а також надає можливість зареєструватися та зберегти персональний комп'ютер до власного профілю. Вебдодаток ділиться на три частини: клієнт, сервер та база даних. Всі вище перераховані модулі розгорнуті у хмарі.

## Перелік умовних позначень

CPU – Central Processing Unit (з англ. – центральний процесор)

GPU — Graphic Processing Unit (з англ. – графічний процесор)

RAM – Random Access Memory (з англ. – оперативна пам'ять)

MB – Motherboard (з англ. – материнська плата)

HDD – Hard Device Drive (з англ. – жорсткий диск)

SDD - Solid-State Drive (з англ. – твердотілий накопичувач)

PS – Power Supply (з англ. – блок живлення)

МГц – мега герц

В – вольт

БД – база даних

## ВСТУП

### Актуальність, наукове та практичне значення обраної теми

На сьогодні комп'ютери займають більшу частину нашого життя. Вони використовуються всюди: у бізнесі, державних структурах, майже у кожного школяра, студента є комп'ютер. Дуже часто задачі, які потребують вирішення, вимагають велику обчислювальну здатність. Наприклад, відео, аудіо, 3-Д обробка, анімація. Програми які цим займаються потребують нових та дорогих компонентів, таких як центральний процесор, відеокарта, оперативна пам'ять, накопичувач, материнська плата та блок живлення. Не завжди комп'ютеру потрібні саме всі дорогі й продуктивні запчастини. Але, наприклад відеоігри є вимогливими до всіх компонентів комп'ютера. І припустимо без відеокарти ви не запустите жодну сучасну гру або ж обробник відео.

Зібрати комп'ютер самому без жодних знань не вийде, а обирати ноутбук не є найкращою ідеєю, бо вони йдуть уже скомплектовані й часто з націнкою, звичайно великим плюсом є мобільність, але інколи вона не потрібна. Також є вже готові збірки від різних магазинів. Але якщо взяти будь-яку таку збірку і перевірити ціну кожної запчастини та порівняти, то ви побачите немало різницю. Багато людей не хочуть просто так переплачувати йц тому вони збирають персональний комп'ютер своїми руками.

Якщо зайти на сторінку[1] з комп'ютерною компонентою, то можна побачити величезну кількість коментарів, таких як: “А чи підійде ця відеокарта \_\_\_\_ до моєї материнської плати \_\_\_\_?” або “Чи взагалі будуть працювати мої комплектуючі з цим процесором \_\_\_\_ ?”.

Щоб розібратися у цій темі людина має витратити немало сил та часу, але він є не у всіх. Ми дуже ледачі створіння, які постійно все спрощують, тож розв'язання такої проблеми, як збірка персонального комп'ютера, не повинна віднімати багато часу.

Отже, підбір комп'ютерних комплектуючих є дуже актуальною проблемою, і щоб розібратися у цьому треба передивитися купу онлайн матеріалу. Сервіс, який перевіряє сумісність деталей та пропонує можливі варіанти є непоганим вирішенням.

### **Мета та завдання курсової роботи**

Мета: Створити зручний вебдодаток для підбору комп'ютерних комплектуючих з можливістю зберігання власних збірок.

Завдання: Розробити вебінтерфейс для користувача, створити базу даних для зберігання даних про комп'ютерні комплектуючі, створити сервер, який оброблює запити вебдодатку та віддає рішення. Розгорнути всі вищезазначені системи в хмару.

### **Об'єкт дослідження**

Наявні труднощі, які виникають у людей, що планують зібрати власний персональний комп'ютер.

### **Використане програмне забезпечення**



Розробка онлайн-додатку для підбору комп'ютерних комплектуючих здійснюється за допомогою об'єктноорієнтованих мов програмування на основі клієнт-серверної архітектури. Створення бази даних впроваджено за допомогою плагіну в IntelliJ Idea для адміністрування реляційних СКБД(Postgresql), обмін інформацією між сервером і клієнтом здійснений за допомогою HTTP запитів на основі RESTful api[2]. Клієнтська частина реалізований за допомогою мови TypeScript та фреймворку Angular. Серверна частина реалізована на мові Java на фреймворку Spring.

### **Джерела дослідження**

Курсова робота зроблена за допомогою знань отриманих з пройдених онлайн курсів по веброзробці на Java та TypeScript, та предметів з університетської програми. Також були дослідження і проаналізовані схожі вебсервіси.

### **Структура роботи**

#### **Робота складається трьох розділів.**

Перший розділ присвячений огляду теми, та обґрунтуванню її актуальності. В роботі є аналіз вже чинних сервісів для допомоги у збірці персонального комп'ютера. Закінчення розділу присвячене постановці задачі, яку треба вирішити.

Другий розділ присвячений обґрунтуванню вибору засобів розробки, та аналіз інших вебдодатків, їх інструментарію для розробки. Також в цьому

розділі аналізуються системи керування базами даних, мови програмування та технології обміну даними між клієнтом і сервером.

Останній розділ обраним алгоритмам для реалізації поставленої задачі. Розділ має технічне завдання, опис всіх можливих функцій для користувача, інструкція у використанні. Обґрунтовано обрані методи розробки. Описані головні модулі сервісу.

## **РОЗДІЛ 1: Аналіз предметної області**

### **1.1 Обґрунтування актуальності задачі**

На сьогодні, майже кожна людина у світі використовує комп'ютер. Але комп'ютер важливий тим, що за його допомогою можна користуватися інтернетом, тобто обмінюватися інформацією зі всією планетою. Статистика з сайту FirstGuide[3] показує, що у світі кількість вебсайтів понад 1.84 мільярди, це не рахуючи закриті від загального доступу ресурсів. Також статистика показує, що інтернет використовується переважною кількістю комп'ютерів у вигляді 54 відсотків, тому вони є невіддільною частиною людства. Для того, щоб обрати комп'ютер треба читати багато літератури. Але, наприклад, для того, щоб комусь подзвонити нам не обов'язково треба розбиратися як працюють телекомунікації. Тож проблема із вибором комп'ютера також не повинна вимагати від нас глибокого розуміння всіх процесів.

### **1.2 Огляд схожих додатків по підбору комп'ютерних комплектуючих**

pcpartpicker [4] – достатньо популярний сайт для підбору комп'ютерних комплектуючих, в нього є зручний інтерфейс, де можна обирати усі можливі комплектуючі, фільтрувати їх, порівнювати різні збірки. Також база даних цих комплектуючих зв'язана з Amazon[5]. Але в такого сайту є недоліки. Наприклад, коли обираєш один компонентів і хочеш пошукати якісь ще, то автофільтрування заважає це зробити. Також, якщо користувач не знає що йому обрати, а більшість користувачів хочуть швидко отримати готове рішення, він не може просто натиснути одну кнопку, для того, щоб йому видало варіанти, які вже відфільтровані за ціною та сумісністю.

buildmurs [6] – ще один сервіс для допомоги по підбору комп'ютерних комплектуючих. Також база даних зв'язана з Amazon[5]. Але функціонал дуже скупий. При виборі комплектуючої відбувається перехід до нової сторінки, це займає багато часу при підборі. Перевірити працездатність обраних компонентів можна тільки при наявності всіх заповнених форм. Немає функції, яка б запропонували недосвідченому користувачу сумісні деталі, до вже обраних. Сам сервіс повільний і має незручний інтерфейс. На цьому сайті не має можливості зберегти свою готову збірку, для майбутньої модифікації.

Lenovo[7] - дуже зручний інструмент для збірки власного ноутбука. Вони навіть дають знижки, якщо ти збереш свій власний ноутбук. Але збірка ноутбуків є обмеженою у виборі комплектуючих, через те що ноутбуки малі. Можливість засунути в них новий потужний процесор є, але потрібні потужні кулери та великі за габаритами радіатори, які в ноутбук не помістяться, а робити більший корпус немає сенсу, бо його вага значно збільшиться і перевага у вигляді мобільності майже зникне. Вага таких ноутбуків понад 10 кілограмів.

Найбільший недолік всіх цих сервісів – це відсутність швидкого способу отримати результат. Треба лізти в різні каталоги, намагатися влучити в сумісний компонент, обирати нові запчастини через окремі вікна. Все це забирає час, який для нас важливий.

### **1.3 Опис вимог до розробки додатку**

Функції, які мають бути реалізовані у вебсервісі:

1. Розробити онлайн-сервіс, який буде мати можливість швидкого та зручного вибору компонентів.
2. Впровадити перевірку на сумісність всіх компонентів у будь-якій комбінації.
3. Додати швидку та інтуїтивно зрозумілу функцію пропозиції компонентів до вже існуючих. При використанні функції мати можливість перевірити сумісність запчастин.
4. Можливість реєстрації для збереження збірок та їх перегляду. При спробі зберегти збірку не увійшовши у профіль, користувач повинен бути перенаправлений на сторінку з авторизацією.
5. Можливість відновити пароль при його втраті. На пошту, по якій зареєстрований користувач, буде надіслано лист з посиланням на відновлення пароля.
6. Можливість переглянути каталог усіх компонентів та побачити усі детальні характеристики.
7. Додати пошук та перегляд всіх характеристик компонентів.
8. Можливість перейти зі сторінки зі збереженими комп'ютерами на сторінку з кожною запчастиною.

#### **1.4 Висновок до розділу 1**

Отже, після аналізу схожих додатків було зрозуміло, що всі вони мають багато незручностей, які треба не допустити у курсовому проєкті. Але й присутні сервіси, які надали корисні ідеї до покращення основного функціонала.

## РОЗДІЛ 2: Теоретичні відомості

### 2.1. Компоненти комп'ютера та їх сумісність

В комп'ютері є 6 основних деталей: процесор (CPU), відеокарта (GPU), материнська плата (Motherboard), блок живлення (Power Supply), оперативна пам'ять (RAM), HDD (жорсткий диск) та SSD (твердотілий накопичувач).

Кожна з них відповідає за окремий функціонал, але всі вони дуже тісно зв'язані одне з одним.

Відеокарта відповідає за обробку та вивід зображення на екран, тобто без відеокарти неможливо побачити картинку. Але є виключення, в сучасні процесори додають графічне ядро і під'єднати дріт можна напряму у материнську плату. Картинка буде, але запустити щось дуже тяжке, наприклад, сучасну відеогру або програму яка оброблює 3-Д об'єкти не вийде. Іноді люди купляють комп'ютери без відеокарт, дуже часто ноутбуки продаються тільки з графічним ядром у процесорі, бо цілю використання ноутбука буде лише робота з браузером, документами, різними десктопними додатками, які не навантажують графічне ядро. Але думка про те, що відеокарта працює лише тоді, коли потрібна обробка 3-Д об'єктів хибна. По перше, це пристрій, який має величезну обчислювальну здатність, і навіть коли ви не маєте справ з дуже тяжким відображенням зображення, то вона бере на себе деяку частину обчислень з процесора. Дискретна відеокарта[8] має графічний процесор, відеопам'ять, цифровий-аналоговий перетворювач, відеоконтролер, постійний запам'ятовувальний пристрій та систему охолодження. Тобто це майже ціла окрема система.

На сьогодні є дві основні компанії на ринку: Nvidia[13] та Amd[11]. Вони постійно конкурують одне з одним і загалом вважається, що Amd

мають дешевші відеокарти з такою ж обчислювальною здатністю, як і у Nvidia, але це стосується лише середнього та низького цінового сегмента. Найновіші та революційні відеокарти з RTX[14] графікою зараз виробляють Nvidia, хоча і дуже дорого.

Окремим групою користувачів відеокарт є люди, які майнять біткоїн[9], вони використовують деяке програмне забезпечення, яке застосовує всі обчислювальні потужності для деяких транзакцій, які додаються в блокчейн. В обмін на це власники отримують криптовалюту. Через цей бум майнінгу зараз дуже великий дефіцит відеокарт на ринку, їх майже нема.

Відеокарта можна обрати на основі відеопам'яті, існує декілька видів: DDR, DDR2, DDR3, GDDR3, GDDR5. Другим фактором є характеристики відео ядра, основною є швидкість, яка вимірюється в мегагерцах. Третій фактор – розрядність шини відеопам'яті, вона вимірюється в бітах. Класифікація за ціновим сегментом:

- Бюджетний сегмент – 64, 128
- Середній сегмент – 128, але іноді й 256
- Дорогий сегмент – 256, 384, 448, 786 і т.д.

Оперативна пам'ять[10] – високошвидкісний компонент в комп'ютері, який тимчасово зберігає інформацію необхідну для обробки всіх процесів, тобто до моменту вимкнення комп'ютера. RAM – це тип пам'яті до якої можна отримати доступ торкаючись попередніх байтів. Оперативна пам'ять тісно зв'язана з процесором, бо вони постійно обмінюються даними одне з одним. На сьогодні основні типи пам'яті це – DDR3 та DDR4. DDR4 – це

новіша технологія, яка має діапазон швидкостей від 1600 МГц до 4800 МГц, це набагато більше ніж у DDR3, від 800 МГц до 3200 МГц. Також новіша пам'ять функціонує на середній напрузі 1.2В на відмінно від DDR3 з 1.4В.

Оперативну пам'ять можна обрати на основі швидкості, яка вимірюється в мегагерцах, типу відеопам'яті, основні: DDR1, DDR2, DDR3, DDR4.

Центральний процесор[13] – це мозок комп'ютера де відбувається більшість обчислень, він є найважливішим елементом комп'ютерної системи. Він має у блоку арифметичної логіки (ALU), який вміє додавати цифри. За допомогою комбінованого блоку управління та ALU центральний процесор може обробляти набагато складніші програми, ніж простий калькулятор. На апаратному рівні центральний процесор - це інтегральна схема, також відома як мікросхема. Інтегральна схема "інтегрує" мільйони або мільярди крихітних електричних деталей, розташовуючи їх у ланцюгах і вміщуючи їх у компактну коробку. На сьогодні найпопулярнішими брендами є Amd[11] та його найновіші моделі Ryzen 7 та Ryzen 9 та Intel[12] з його Intel Core i9 10-го покоління.

Щоб обрати процесор, треба дивитися на його покоління, кількість ядер та потоків на кожному ядрі. Технологія hyperthreading[13] дозволяє обробляти на кожному ядрі по декілька потоків, що значно прискорює обрахунки. Також важливим фактором є рівень кешу, він дозволяє прискорити отримання даних. Існує 3 види:

L1 – швидкий, але малий.

L2 – більший ніж L1, але також і повільніший.



L3 – дуже просторий, але не є найшвидшим.

Також дуже важливою є швидкість ядра, яка вимірюється в ГГц. Найновіші процесори мають швидкість 3.7 – 4.2 ГГц.

Материнська плата – це пристрій, який зв’язує абсолютно всі компоненти до одної цілої системи. Вона складається з великої кількості запчастин.

Основні:

- Місце для процесора, яке має назву “Cpu Socket”.
- Місце для відеокарти, яке має назву “PCI Express”.
- Всі можливі конектори для кулерів.
- Місце для оперативної пам’яті, яке має свій тип: DDR1, DDR2, DDR3, DDR4.
- USB, відео, аудіо, інтернет входи

У материнської плати є різні формфактори: AT, ATX, Baby AT, BTX, DTX, microATX... Основні з них це ATX та microATX, вони відрізняються розміром та кількістю конекторів. Наприклад, в microATX не можна вставити дві відеокарти, менша кількість місць для оперативних пам’яті й т.д. Але не всім людям потрібен величезний комп’ютерний кейс, тож вони збирають комп’ютер на основі microATX. Також на сьогодні, важливим фактором є присутність m2 входу на материнській платі. M2 – це твердотілий накопичувач, який є найшвидшим серед своїх родичів.

Для того, щоб обрати материнську плату, треба дивитися її максимальну швидкість для оперативної пам’яті, сокет для процесора,

підтримку M2, максимальну оперативну пам'ять у гігабайтах, максимальну кількість плашок оперативної пам'яті.

Блок живлення [17] – компонент, який подає живлення на компоненти комп'ютера. Він отримує живлення від розетки й перетворює струм зі змінного в постійний, що вимагає комп'ютер. PS виконує задачу по контролю напруги, що в результаті надає комп'ютеру змогу працювати постійно, поки він під'єднаний до мережі. Блок живлення є невіддільною частиною будь-якого комп'ютера і повинен нормально функціонувати, щоб інші компоненти працювали. Єдиний дрiт, який йде до комп'ютера це живлення блоку. Якби ви відключили джерело живлення, це було б схоже на металеву коробку з вентилятором всередині і деякими кабелями, прикріпленими до неї. Хоча більшість комп'ютерів мають внутрішні блоки живлення, багато електронних пристроїв використовують зовнішні. Наприклад, деякі монітори та зовнішні жорсткі диски мають джерела живлення, розташовані поза основним блоком. Ці джерела живлення підключені безпосередньо до кабелю, який підключається до розетки. Вони часто включають інший кабель, який підключає пристрій до джерела живлення. Обидві ці конструкції дозволяють зробити основний пристрій меншим.

PS живить такі компоненти:

- Материнська плата, на материнській платі є вхід, який в основному складається з 24 пінів.
- Процесор через материнську плату, на ній є вхід для живлення процесора.

- Відеокарта, в багатьох нових моделях є додаткове живлення.
- Твердотілі накопичувачі та жорсткі диски.

У блока живлення є максимальна кількість ват. Треба обирати його так, щоб сумарна кількість усіх компонентів комп'ютера, включаючи периферійні пристрої, не перевищувала заявлену кількість на блоці живлення. Також треба дивитися на дроти і їх тип, щоб живлення материнської плати, процесора та відеокарти збігалися.

Оскільки джерело живлення є першим місцем, де електронний пристрій отримує електроенергію, він також є найбільш вразливим до стрибків напруги та стрибків напруги. Отже, джерела живлення призначені для боротьби з коливаннями електричного струму і при цьому забезпечують регульовану або постійну вихідну потужність. Деякі включають запобіжники, які будуть перегорати, якщо перенапруга занадто велика, захищаючи решту обладнання. Адже замінити блок живлення набагато дешевше ніж весь персональний комп'ютер.

Сумісність всіх компонентів комп'ютера дуже важлива, якщо щось не підходить одне до одного, то в кращому випадку комп'ютер може просто не ввімкнутися, а в гіршому щось перегорить.

## 1. Сумісність материнської плати та оперативної пам'яті.

- 1.1. У материнської плати є максимальна кількість плашок оперативної пам'яті, які можна поставити. Завжди треба обирати таку кількість, яка не перевищує допустиму кількість на материнській платі.

- 1.2. Також є обмеження в вигляді максимальної кількості гігабайт оперативної пам'яті. Завжди треба обирати кількість гігабайт меншу або рівну тій, що на материнці.
- 1.3. Третя характеристика це тип оперативної пам'яті, треба, щоб вони збігалися. Наприклад пам'ять DDR4 працює лише с типом DDR4 на материнській платі.
2. Сумісність материнської плати та твердотілого накопичувача M2.  
Не на всіх моделях є роз'єм для M2, тож дуже уважно треба дивитися на характеристики, в них завжди це пишеться.
3. Сумісність процесора та материнської плати.  
У процесора є роз'єм, він зветься сокетом, сокет на платі та процесорі повинні збігатися один в один.
4. Сумісність блока живлення та материнської плати.
  - 4.1. У блока живлення є піни, які можуть підходити під декілька плат. В основному це 20+4 pins. Дріт з блоку живлення повинен завжди заповнювати всі піни на материнській платі.
  - 4.2. У блока живлення є піни для процесора, які можуть підходити під декілька плат. В основному це 4+4 pins. Дріт з блоку живлення повинен завжди заповнювати всі процесорні піни на материнській платі.
  - 4.3. У блока живлення є піни для відеокарти, які можуть підходити під декілька плат. В основному це 4\*6+2 pins. Дріт з блоку живлення повинен завжди заповнювати всі піни на відеокарті.

## 2.2 Типи комп'ютерних збірок

Офіційної класифікації комп'ютерних збірок немає, але загалом можна визначити кілька типів:

- офісний комп'ютер – збірка, яка має середню ціну, дуже часто не має відеокарти, бо вони не потрібні для роботи з документами та браузером. Але повинна мати непоганий процесор та бажано SSD накопичувач, він сильно прискорює роботу комп'ютера.

- серверний комп'ютер – використовується в основному віддалено, для обробки великої кількості даних, повинен мати дуже потужний процесор. Через те, що спілкування з сервером відбувається завдяки консолі, він не потребує відеокарти.

- комп'ютер для 3-Д обробки, анімацій та ігор – завжди повинен мати відеокарту, бо без неї жодна із спеціалізованих програм або ігор може навіть не запуститися. Також дуже велику роль грають оперативна пам'ять та процесор. Рекомендується мати SSD накопичувач.

## 2.3 Переваги обраної СКБД та сервісу де розгорнутий додаток

Postgresql[19] – СКБД, яка підтримує велику кількість мов програмування, які можуть викликати функцій: SQL, Java, Javascript, Python, Lua, R, Perl, Tcl...

Одним із найбільш фундаментальних способів, яким PostgreSQL відрізняється від більшості інших реляційних баз даних, є його основна конструкція.

Більшість реляційних баз даних найкраще описати як Реляційні системи управління базами даних (СУБД). СУБД - це програмне забезпечення, спеціально розроблене для обробки реляційних баз даних, де дані зберігаються у структурах, подібних до таблиць, із заздалегідь визначеними стовпцями та типами даних. Дані можна запитувати, модифікувати та отримувати, використовуючи методи, засновані на реляційній алгебрі, зазвичай за допомогою мови структурованих запитів (SQL).

PostgreSQL, навпаки, технічно є об'єктно-реляційною системою управління базами даних (ORDBMS). Це означає, що він має ті самі реляційні можливості, що і СУБД, але додатково має деякі об'єктоорієнтовані функції.

Postgresql має такі можливості:

- Визначати власні складні типи даних.
- Функції перевантаження для роботи з різними типами даних аргументів.
- Визначати відносини успадкування між таблицями.

Ці функції є потужними інструментами, які допомагають працювати з базами даних та даними, використовуючи ті самі методи, з якими ви могли бути знайомі під час програмування. Підвищена гнучкість дозволяє моделювати різні типи та взаємозв'язки в системі баз даних, а не зовні в програмах. Це може допомогти зберегти узгодженість та нав'язати передбачувану поведінку ближче до фактичних даних.

Також ця СКБД обрана через те, що її дуже легко розгортати через хмарний сервіс Heroku[18].

Heroku – це хмарна платформа, яка дозволяє компаніям створювати, доставляти, контролювати та масштабувати додатки. Є можливість

розгорнути свій додаток, в безкоштовному пакеті можна отримати 550 годин роботи серверу на місяць, є можливість зв'язати базу даних з додатком. Всього безкоштовно можливо мати на одному акаунті 4 сервіси.

Тому для курсового проєкту це ідеальний сервіс для розгортання всіх модулів моєї системи.

## **2.4 Висновок до розділу 2**

Отже, в цьому розділі було проаналізовано велику кількість теорії про комп'ютерну техніку. Всі можливі сумісності компонентів та їх невід'ємність одне від одного. Також було визначено різні типи збірок: офісна, серверна, для обробки 3-Д об'єктів. Було визначено переваги СКБД та обраного хмарного сервісу.

## РОЗДІЛ 3: Проектування та розробка додатку

### 3.1 Опис технічного завдання на розробку

Технічне завдання на тему “Розробка вебдодатку для підбору комп'ютерних комплектуючих”.

Користувач повинен мати такі можливості:

- Пошук за ім'ям таких видів деталей: GPU (відеокарта), MB (материнська плата), PS (блок живлення), RAM (оперативна пам'ять), CPU (центральний процесор). Можливість вибору цих деталей для майбутнього аналізу.
- Перевірка сумісності обраних деталей. При обиранні не всіх комплектуючих, перевіряються лише ті деталі на сумісність що є.
- Пропозиція нових деталей, які можна поставити до раніше обраних. При несумісності обраних деталей, видавати всю можливу інформацію з причинами.
- Можливість полагодити несумісну збірку деталями що існують, які б підходили до поточної комбінації.
- Можливість зареєструватися на сервісі за допомогою логіну у вигляді email, та пароля, який повинен бути не менше 8 символів.
- Можливість авторизуватися на сервісі за допомогою логіну та пароллю.
- Можливість відновити пароль за допомогою пошти. На неї приходить лист із посиланням на відновлення пароля.



- Можливість зберегти збірку у свій профіль. При спробі зберегти збірку неавторизованому користувачеві, його буде перенаправлено на сторінку з авторизацією.
- Можливість переглянути або видалити збірку у власному кабінеті.
- Можливість перейти на кожну окрему деталь із власних збірок через посилання у таблиці.
- Можливість детально переглянути кожний компонент окремо перейшовши на сторінку “Каталог”.

Скористатися сервісом можна за такою схемою:

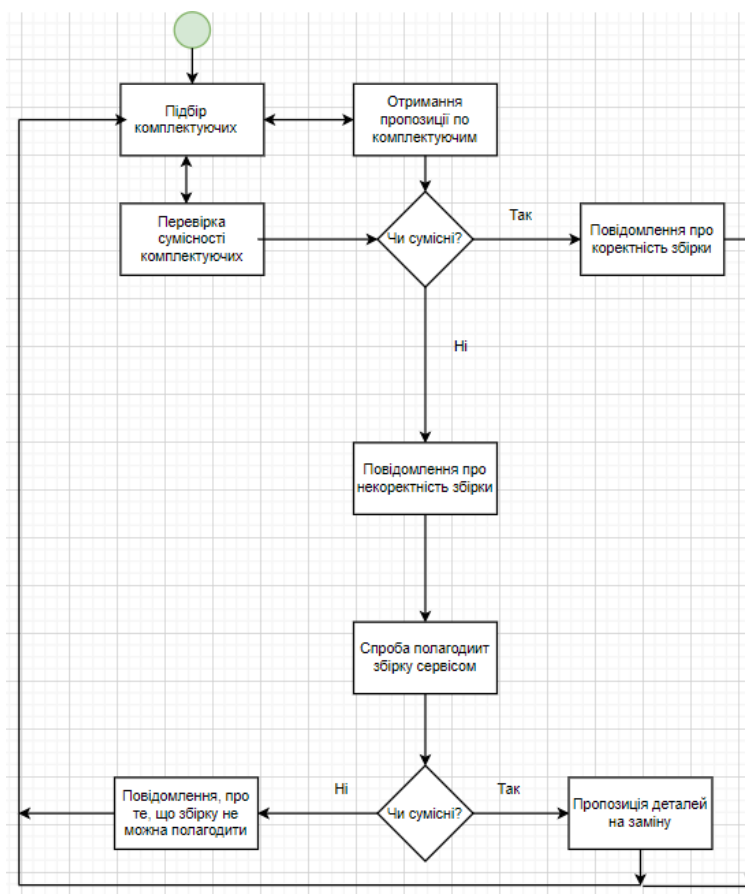


Рисунок 3.1.1 - Схема роботи основних функцій

Скористатися можливістю збереження збірки можна за такою схемою:



Рисунок 3.1.2 - Схема авторизації та збереження власних збірок

В основі сервісу лежить взаємодія клієнту з сервером на основі REST API. Сервер зв'язується з базою даних, яка оброблює дані користувача, а також зберігає інформацію про збірки.

Сервіс може бути використаний будь-ким у кого є інтернет, адже для роботи потрібен лише браузер. Через те, що інтерфейс легкий та зручний, користувач без навичок може легко розібратися у роботі додатку.

### 3.2 Обґрунтування обраних засобів розробки

Розробка вебсервісу залежить від глибокого розуміння роботи кожного модуля додатку.

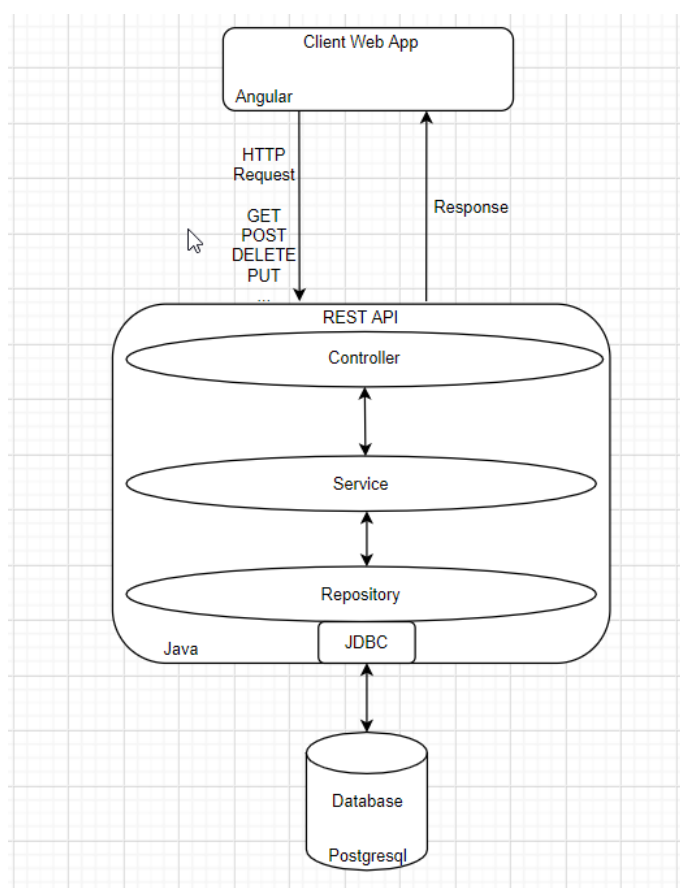


Рисунок 3.2.1 – Архітектура додатку

Клієнт-серверна архітектура має багато клієнтів, які надсилають запити й отримують відповідь від сервера, який все це паралельно оброблює. Клієнтські комп'ютери забезпечують відображення інтерфейсу. Сервер повинен надавати прозорий інтерфейс, який має бути інтуїтивно зрозумілий. Клієнт у випадку комп'ютерного сервісу це браузер, який знаходиться на кінцевих девайсах у користувачів. Сервер знаходиться в іншому місці мережі на більш потужній машині. Наприклад, при обробці даних у нашому сервісі, на клієнтському комп'ютері може бути запущений браузер для введення інформації про комп'ютер, тоді як на сервері працює інша програма, яка управляє базою даних, в якій інформація зберігається постійно. Багато клієнтів можуть одночасно отримувати доступ до інформації сервера, а одночасно клієнтський комп'ютер може виконувати інші завдання. Оскільки й клієнтський, і серверний комп'ютери вважаються незалежними пристроями, модель клієнт-сервер повністю відрізняється від старої моделі мейнфреймів, в якій централізований мейнфреймовий комп'ютер виконував усі завдання для пов'язаних з ним "німих" терміналів, які просто спілкувалися з центральною мейнфреймом. Це дуже неефективна архітектура, бо всі обробки відбуваються на стороні клієнта, а це потребує потужних комп'ютерів, завантаження та установку програмного забезпечення.

Клієнт та сервер обмінюються інформацією на основі RESTful API[21], що розшифровується як Representational State Transfer, в перекладі з англійської - передача стану представлення. Цей архітектурний стиль API, який використовує запити HTTP[20] для доступу та використання даних. Ці дані можуть бути використані такими командами: GET (отримання), POST(встановлення), PUT (оновлення) та DELETE (видалення). Ця

архітектура є стилем та підходом до комунікацій, часто використовуваних при розробці вебслужб.

Зазвичай технологія REST є кращою перед іншими подібними технологіями. Це відбувається, оскільки REST використовує меншу пропускну здатність, що робить його більш придатним для ефективного використання Інтернету.

На сервері використовується шаблон проєктування MVC[22], тобто Model, View, Controller. Але у випадку REST арі, замість View ми віддаємо JSON[23], тобто деякі оброблені дані у відповідь на запит клієнта.

1. Model – це дані, що використовуються програмою. Це може бути база даних, файл або простий об'єкт, наприклад піктограма чи персонаж у відеоіграх.
2. Controller – оновлює як моделі, так і подання. Він приймає запит користувача та виконує відповідні розрахунки, щоб повернути коректні дані. Контролер приймає введені користувачем дані та відповідно оновлює модель і вигляд.

Також на сервері використовується шаблон Репозиторій-Сервіс.

Контролер – це найвищий рівень, який приймає запити від клієнта, в нього додають залежність сервісу, і він викликає деякі методи на сервісі.

Сервіс – це середній рівень, де виконується вся бізнес-логіка. В ньому знаходиться залежність репозиторію, на якому викликається метод по отриманню даних.

Репозиторій – це найнижчий рівень цієї ієрархії, він працює з базою даних, його задача – отримати дані і відати їх сервісу.

Таким чином кожен вищий рівень потребує наявності залежності нижчого рівня. Через те що ми поділяємо нашу програму на такі 3 рівні, ми можемо чітко відділити бізнес-логіку від всього іншого.

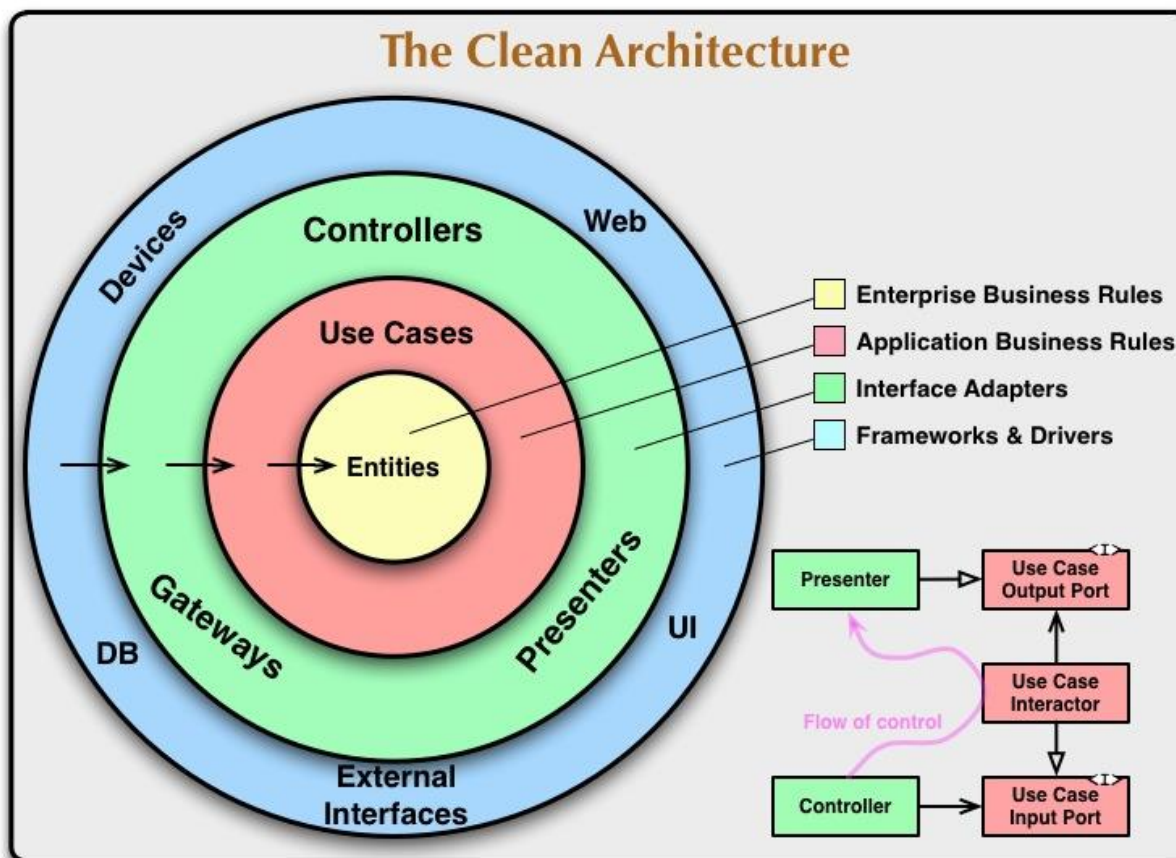


Рисунок 3.2.2 - Схема роботи сервісу [24]

Для цього проекту було реалізовано вище перераховані патерни на мові Java. Оскільки Java допомагає розробляти надійні, масштабовані та багаторівневі додатки для різних бізнес-вимог, багато компаній програмного забезпечення пропонують цілі послуги з розробки програмного забезпечення. Java дозволяє розробляти від невеликих додатків до великих наскрізних корпоративних рішень. До того, на відміну від інших мов програмування,

роль та зручність використання яких зменшилися зі зміною часу та технологій, Java з кожним останнім оновленням стає міцнішою.

Spring - це найпопулярніший фреймворк розробки додатків для Enterprise Java. Мільйони розробників у всьому світі використовують Spring Framework для створення високопродуктивного та багаторазового коду. Spring framework – це Java фреймворк з відкритим кодом. Базова версія Spring framework становить близько 2 МБ. Основні особливості Spring Framework можна використовувати при розробці будь-якої програми Java, але є розширення для побудови вебдодатків поверх платформи Java EE. Ціль Spring – це спростити розробку J2EE[27] та сприяти передовій практиці програмування, включивши модель програмування на основі POJO[25].

Angular - це платформа та фреймворк для побудови односторінкових клієнтських програм за допомогою HTML та TypeScript. Angular написаний на TypeScript. Він реалізує основну та додаткову функціональність як набір бібліотек TypeScript, які ви імпортуєте у свої програми.

Архітектура програми Angular спирається на певні фундаментальні концепції. Основними будівельними елементами Angular framework є компоненти, які організовані в NgModules. NgModules збирають відповідний код у функціональні набори; Додаток Angular визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувати, і, як правило, має набагато більше модулів функцій.

На сьогодні Angular є одним із лідерів серед веброзробки. По статистиці від сайту sam-solutions[27] Angular займає 6 місце серед найпопулярніших фреймворків для веброзробки. Spring займає 5 місце.

JWT[28] – JSON Web Token - це закодований рядок, який захищений URL-адресою, який може містити необмежений обсяг даних (на відміну від файлів cookie), і він підписаний криптографічно.

Коли сервер отримує JWT, він може гарантувати, що дані, які він містить, можуть бути надійними, оскільки вони підписані джерелом. Жоден посередник не може змінити JWT після його надсилання.

Важливо зазначити, що JWT гарантує право власності на дані, але не шифрування; дані JSON, які ви зберігаєте у JWT, може бачити кожен, хто перехоплює маркер, оскільки він просто серіалізований, а не зашифрований.

Тому в цій курсовій роботі обраний JWT для авторизації.

### **3.3 Опис реалізації**

Додаток побудований на основі клієнт-серверної архітектури. Клієнтом є браузер. Клієнтська частина написана за допомогою мови TypeScript та фреймворку Angular. Серверна частина написана на мові Java та використовується фреймворк Spring. Клієнт та сервер обмінюються інформацією за допомогою HTTP запитів, таких як: GET, POST, PUT та DELETE. Сервер пише та читає з бази даних за допомогою СКБД PostgreSQL. Сервер може одночасно обробляти запити від багатьох клієнтів, тому що фреймворк Spring працює у багатопотоковому режимі, використовуючи Java.

### **3.4. Опис реляційної моделі**

База даних складається з 14 сутностей.



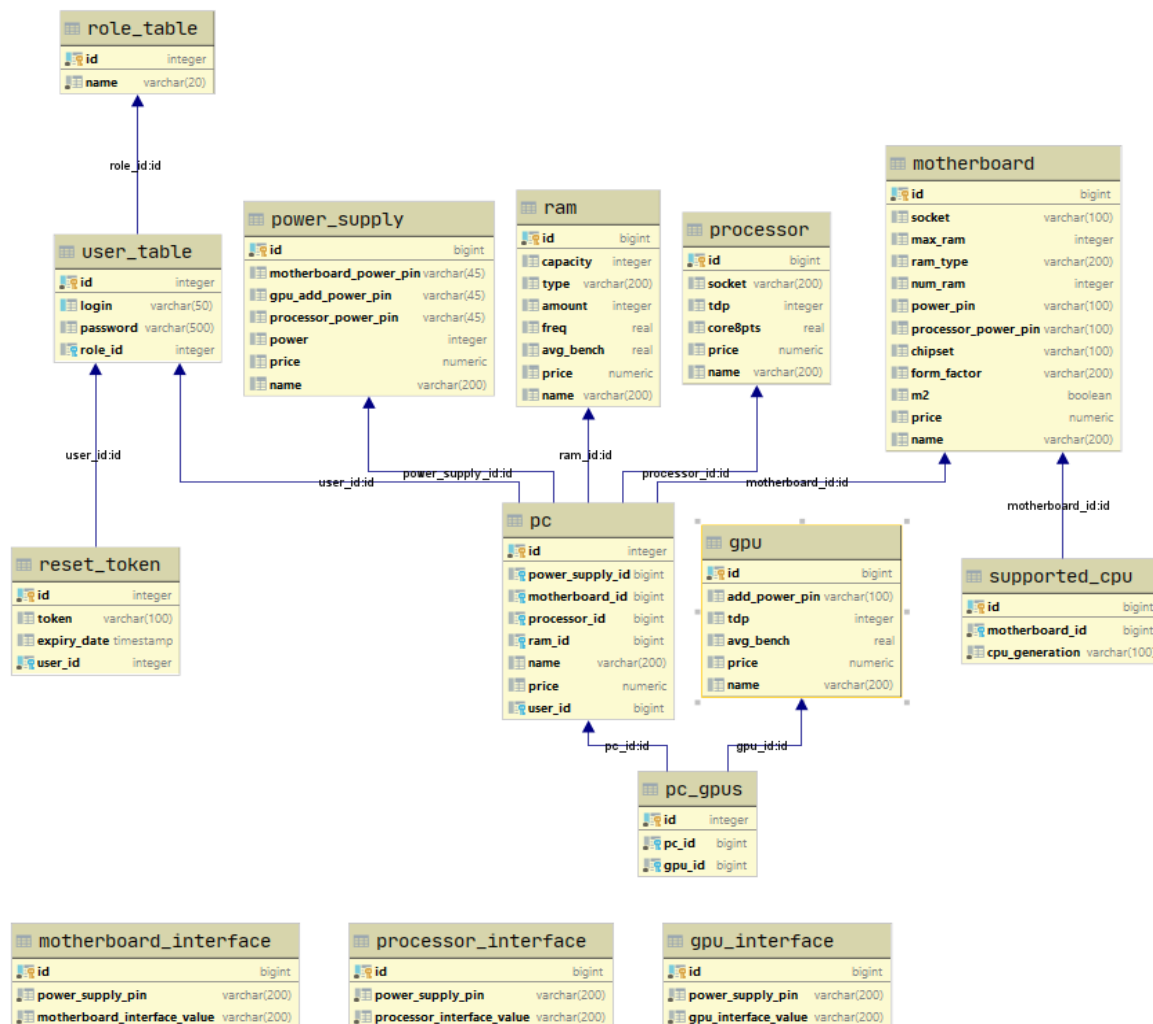


Рисунок 3.4.1 – ER діаграма

Основні таблиці БД:

Сутність “power\_supply” - блок живлення.

Таблиця 3.4.2 – Таблиця “power\_supply”

power_supply				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
РК	id	bigint	NOT NULL	Обов'язковий атрибут для ідентифікації
	motherboard_power_pin	varchar2(100)	NOT NULL	Характеристика дроту для живлення материнської плати
	gpu_add_power_pin	varchar2(100)	NOT NULL	Характеристика дроту для живлення відеокарти
	processor_power_pin	varchar2(100)	NOT NULL	Характеристика дроту для живлення процесора
	power	number	NOT NULL	Потужність блоку живлення
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “pc\_gpus ”, відношення багато до багатьох таблиці “gpu” з “pc”, тому що в одного комп'ютера може бути кілька відеокарт.

Таблиця 3.4.8 – Таблиця “pc\_gpus”

pc_gpus				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
FK	pc_id	bigint	NOT NULL	Зв'язок з таблицею pc: id. Зв'язок обов'язковий
FK	gpu_id	bigint	NOT NULL	Зв'язок з таблицею gpu:id. Зв'язок обов'язковий

Таблиця 3.4.3 – Таблиця “ram”

ram				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	Обов'язковий атрибут для ідентифікації
	capacity	number	NOT NULL	Кількість гігабайт на одній плашці
	type	varchar2(100)	NOT NULL	Тип оперативної пам'яті
	amount	number	NOT NULL	Кількість плашок пам'яті
	freq	real	NOT NULL	Швидкість у МГц
	avg_bench	real	NOT NULL	Ефективність
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “processor” - центральний процесор.

Таблиця 3.4.4 – Таблиця “processor”

processor				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	Обов'язковий атрибут для ідентифікації
	socket	varchar2(100)	NOT NULL	Роз'єм
	tdp	number	NOT NULL	Відведення якої теплової енергії у ватах
	core8pts	real	NOT NULL	Ефективність
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “motherboard” - материнська плата

Таблиця 3.4.5 – Таблиця “motherboard”

motherboard				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	Обов'язковий атрибут для ідентифікації
	socket	varchar2(100)	NOT NULL	Роз'єм для процесора
	max_ram	number	NOT NULL	Максимальна кількість гігабайт оперативної пам'яті
	ram_type	varchar2(100)	NOT NULL	Тип оперативної пам'яті
	num_ram	decimal	NOT NULL	Максимальна кількість плашок оперативної пам'яті
	power_pin	varchar2(100)	NOT NULL	Тип входу для живлення
	processor_power_pin	varchar2(100)	NOT NULL	Тип входу для живлення процесора
	chipset	varchar2(100)	NOT NULL	Чипсет
	form_factor	varchar2(100)	NOT NULL	Формфактор
	m2	boolean	NOT NULL	Підтримка SSD M2
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “gru” - відеокарта

Таблиця 3.4.6 – Таблиця “gru”

gru				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	Обов'язковий атрибут для ідентифікації
	add_power_pin	varchar2(100)	NOT NULL	Роз'єм
	tdp	number	NOT NULL	Відведення якої теплової енергії у ватах
	avg_bench	real	NOT NULL	Ефективність
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “pc”, тобто комп'ютер зв'язує усі компоненти: процесор, материнську плату, оперативну пам'ять, блок живлення. Відношення до всіх інших деталей – один до багатьох. Тобто для одного компонента багато “комп'ютерів” Надалі використовується для перевірки на сумісність, пропозиції компонентів та налагодження збірки.

Зв'язок із таблицею “user\_table”. У кожного користувача може бути багато збережених комп'ютерів.

Таблиця 3.4.7 – Таблиця “pc”

pc				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
FK	power_supply_id	bigint	NULL	Зв'язок з таблицею power_supply: id. Зв'язок необов'язковий
FK	motherboard_id	bigint	NULL	Зв'язок з таблицею motherboard:id. Зв'язок необов'язковий
FK	processor_id	bigint	NULL	Зв'язок з таблицею processor:id. Зв'язок необов'язковий
FK	ram_id	bigint	NULL	Зв'язок з таблицею ram:id. Зв'язок необов'язковий
FK	user_id	bigint	NULL	Зв'язок з таблицею user_table:id. Зв'язок обов'язковий
	price	decimal	NOT NULL	Ціна за одиницю у доларах
	name	varchar2(100)	NOT NULL	Назва

Сутність “user\_table” - користувач. Має атрибут пароль, логін та айді ролі, до якої належить користувач. Зв’язок один до багатьох, тому що у користувача може бути багато ролей.

Таблиця 3.4.9 – Таблиця “user\_table”

user_table				
Ключ	Ім’я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
	login	varchar2(300)	NOT NULL	Логін користувача
	password	varchar2(300)	NOT NULL	Пароль користувача
FK	role_id	bigint	NOT NULL	Зв’язок з таблицею role_table: id. Зв’язок обов’язковий

Сутність “role\_table” – роль користувача.

Таблиця 3.4.10 – Таблиця “role\_table”

role_table				
Ключ	Ім’я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
	name	varchar2(100)	NOT NULL	Назва ролі



Сутність “reset\_token” – токен, який використовується для відновлення паролю, в нього є “expiry\_date” – дата, коли токен стане не валідним.

Токен прив’язаний до конкретного користувача.

Таблиця 3.4.11 – Таблиця “reset\_token”

reset_token				
Ключ	Ім’я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
	token	varchar2(100)	NOT NULL	Токен для відновлення пароля
	expiry_date	date	NOT NULL	Дата коли токен стане не валідним
FK	user_id	bigint	NOT NULL	Зв’язок з таблицею user_table: id. Зв’язок обов’язковий

Також є три таблиці “processor\_interface”, “gpu\_interface”, “motherboard\_interface” для збереження даних про роз’єм та відповідний йому дріт з блоку живлення.

Таблиця 3.4.12 – Таблиця “\*\_interface”

*_interface				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	bigint	NOT NULL	
	power_supply_pin	varchar2(100)	NOT NULL	Дріт з деякою кількістю пінів
	*_interface_value	varchar2(100)	NOT NULL	Сумісний роз'єм

### 3.5 Розробка головного сервісу

Робота сервісу починається з запуску серверу. Сервер встановлює з'єднання та виділяє собі пул з'єднань для доступу до даних. Сервер очікує HTTP запитів. Запускається вебсайт, який стоїть на окремому сервері. Клієнт посилає запит з браузера на сервер, йому приходить відповідь і клієнтська частина, яка написана на Angular, відображує дані на вебсторінці.

Підбір комплектуючих. При вводі значення в поле для пошуку, на кожний символ спрацьовує асинхронна функція, яка посилає запит на сервер для пошуку відповідної деталі за введеною стрічкою, у відповідь приходять перші 10 варіантів, які мають введені символи.

Перевірка сумісності. На сервер надсилаються айді усіх обраних деталей. Сервіс на сервері перевіряє сумісність та повертає JSON об'єкт, який має булеві поля, що кажуть що з чим сумісно, а що ні. Клієнтська частина це обробляє і показує користувачу.

Пропозиція нових компонент. На сервер надсилаються айді усіх обраних користувачем компонентів. Сервіс спочатку перевіряє на сумісність всі компоненти користувача, а потім пропонує найкращі.

Виправлення несумісної збірки відбувається на основі материнської плати, вона є ключовим елементом. Коли усім несумісним деталям знайдено заміну, сервер відповідає користувачу.

Реєстрація. Користувач вводить пароль та логін, які валідує клієнтська частина на те, щоб email підпадав під такий regex: “`^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$`”, а пароль мав кількість символів не менше ніж 8. Потім надсилається запит на сервер з логіном та паролем. Сервер валідує дані. Спочатку йде пошук користувача з таким же логіном, якщо таких не має, пароль хешується з сіллю за допомогою хеш алгоритму, після всіх цих процесів створюється новий запис в базі даних.

Авторизація. Користувач вводить свій логін та пароль та надсилає запит. Сервіс шукає юзера з таким логіном, якщо такий користувач існує він хешує з сіллю переданий пароль та порівнює його з хешованим паролем з бази даних. Якщо вони збігаються, генерується новий JWT, який повертається клієнту. JWT стає не валідним, якщо ним не користуватися 3 години. У JWT є логін користувача, завдяки якому клієнтська частина може створити сесію.

Відновлення пароля. Користувач вводить пошту по якій він був зареєстрований та надсилає запит на сервер. Сервіс перевіряє чи існує така пошта. Якщо вона існує, то генерується новий рандомний токен, який зберігається в базу даних для цього користувача, та надсилається посилання(н.п. - <https://computerservice.herokuapp.com/save-password?token=33281f65->

[5fb6-46d8-9acf-91db9fd964f7](#)) на пошту юзера, де параметром в посиланні є згенерований токен. Користувач переходить по посиланню та вводить новий пароль, який повинен відрізнятися від попереднього. На сервері перевіряється токен з посилання з токеном в базі. Якщо вони збігаються новий пароль зберігається для користувача.

### 3.6 Тестування програми і результати її виконання

Переходимо на головну сторінку <https://computerservice.herokuapp.com>.

Бачимо головну сторінку з формою для пошуку компонентів. Якщо обрати компонент, то максимальна ціна блокується, тому що для цього компоненту вже не має сенсу обирати максимальну ціну. Кнопка “CHECK COMPATIBILITY” неактивна, бо треба вибрати хоча б дві деталі. Кнопка “PROPOSE COMPONENTS” дезактивується, якщо всі елементи обрані.

На верхньому барі можна побачити 4 відділи. “Home” – головна сторінка на якій ми зараз знаходимося. “Catalog” – сторінка де можна побачити всі компоненти з їх характеристиками. “Saved PC-s” – сторінка зі збереженими комп’ютерами, доступна лише авторизованому користувачу. “Log in” – сторінка з авторизацією.

Home Catalog Saved PC-s Log in

GPU:  GPU Max Price: \$ 10000 .00

Motherboard:  Motherboard Max Price: \$ 10000 .00

RAM:  RAM Max Price: \$ 10000 .00

Power supply:  Power Supply Max Price: \$ 10000 .00

CPU:  Processor Max Price: \$ 10000 .00

CHECK COMPATIBILITY PROPOSE COMPONENTS

Рисунок 3.6.1 – Головна сторінка вебсайту

Обираємо компоненти через select – search форму та натискаємо на кнопку “CHECK COMPATIBILITY”. Бачимо повідомлення, що в нас несумісний сокет процесора з материнською платою.

Home Catalog Saved PC-s Log in

GPU:  GPU Max Price: \$ 10000 .00

Motherboard:  Motherboard Max Price: \$ 10000 .00

RAM:  RAM Max Price: \$ 10000 .00

Power supply:  Power Supply Max Price: \$ 10000 .00

CPU:  The processor socket is not compatible with a motherboard socket. Processor Max Price: \$ 10000 .00

CHECK COMPATIBILITY PROPOSE COMPONENTS FIX

Рисунок 3.6.2 – Функція перевірки компонентів на сумісність

Для таких випадків можна обрати інші компоненти, чи скористатися функцією fix, яка знайде нам сумісні деталі. Натискаємо на кнопку “FIX”, та отримуємо в результат процесори зі схожими характеристиками, але лише сумісні.

Home Catalog Saved PC's Log in

GPU: GTX 1070-Ti GPU Max Price: \$ 10000 .00

Motherboard: MSI MAG Z390 Motherboard Max Price: \$ 10000 .00

RAM: G.SKILL Flare 16GB RAM Max Price: \$ 10000 .00

Power supply: Power supply search... Power Supply Max Price: \$ 10000 .00

CPU: i7 8700k Processor Max Price: \$ 10000 .00

We have found few processors that you could pick so that your PC assembly will work!

Name: i5-9600k | Price: \$219.00 | [Click me to select this processor](#)

Name: i5-8600K | Price: \$240.00 | [Click me to select this processor](#)

Name: i5-8500 | Price: \$275.00 | [Click me to select this processor](#)

Name: Xeon E3-1240 | Price: \$230.00 | [Click me to select this processor](#)

Рисунок 3.6.3 – Функція виправлення несумісної збірки.

Обираємо будь-який з них, та пробуємо натиснути на кнопку “CHECK COMPATIBILITY” ще раз. Бачимо, що наша збірка тепер працює.

Home Catalog Saved PC's Log in

GPU:  GPU Max Price:  \$ 10000 .00

Motherboard:  Motherboard Max Price:  \$ 10000 .00

RAM:  RAM Max Price:  \$ 10000 .00

Power supply:  Power Supply Max Price:  \$ 10000 .00

CPU:  Processor Max Price:  \$ 10000 .00

This PC assembly will work!

[CHECK COMPATIBILITY](#) [PROPOSE COMPONENTS](#) [SAVE](#)

Рисунок 3.6.4 – Перевірка сумісності на новій комбінації

Спробуємо обрати два компоненти, для інших обираємо максимальну ціну, натискаємо на кнопку “PROPOSE COMPONENTS”. Бачимо, що нам запропонували сумісні та найкращі компоненти для цієї комбінації.

Home Catalog Saved PC's Log in

GPU:  GPU Max Price:  \$ 10000 .00

Motherboard:  Motherboard Max Price:  \$ 873 .00

We have found few motherboards that you could pick so that your PC assembly will work:  
Name: Asus WS X299 | Price: \$400.00 | [Click me to select this motherboard](#)

RAM:  RAM Max Price:  \$ 498 .00

We have found few RAM's that you could pick so that your PC assembly will work:  
Name: G.SKILL Flare 16GB | Price: \$318.00 | [Click me to select this RAM](#)  
Name: Vengeance LPX 32GB | Price: \$184.00 | [Click me to select this RAM](#)  
Name: Corsair Vengeance RGB PRO 32GB | Price: \$165.00 | [Click me to select this RAM](#)  
Name: Trident Z C16 16GB | Price: \$99.00 | [Click me to select this RAM](#)  
Name: Ripjaws 4 C15 16GB | Price: \$92.00 | [Click me to select this RAM](#)

Power supply:  Power Supply Max Price:  \$ 10000 .00

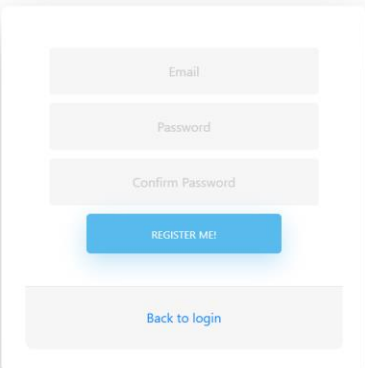
CPU:  Processor Max Price:  \$ 348 .00

We have found few processors that you could pick so that your PC assembly will work:  
Name: i5 9600K | Price: \$219.00 | [Click me to select this processor](#)

[CHECK COMPATIBILITY](#) [PROPOSE COMPONENTS](#)

Рисунок 3.6.5 – Перевірка функції пропозиції компонентів

Реєструємося за допомогою логіну та пароля. Якщо все ввести правильно нас перенаправить на сторінку з авторизацією

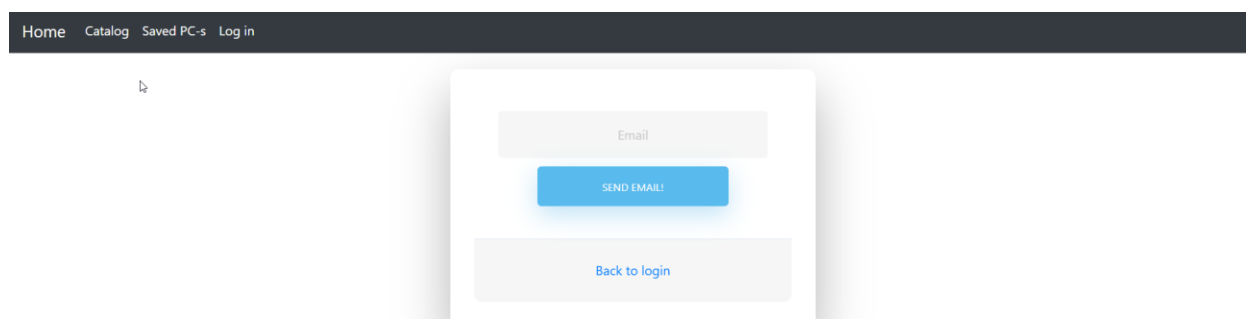


The image shows a registration form centered on a dark gray navigation bar. The navigation bar contains links: Home, Catalog, Saved PC-s, and Log in. The registration form is a white box with a light gray border. It contains three input fields for Email, Password, and Confirm Password, each with a light gray placeholder text. Below these fields is a blue button labeled REGISTER ME!. At the bottom of the form is a light gray button labeled Back to login.

Рисунок 3.6.6 – Реєстрація

Припустимо ми забули пароль. Натискаємо на кнопку “Forgot Password?”. Вводимо свій email. На пошту повинен прийти лист з посиланням (н.п. - <https://computerservice.herokuapp.com/save-password?token=33281f65-5fb6-46d8-9acf-91db9fd964f7>). Переходимо по ній.

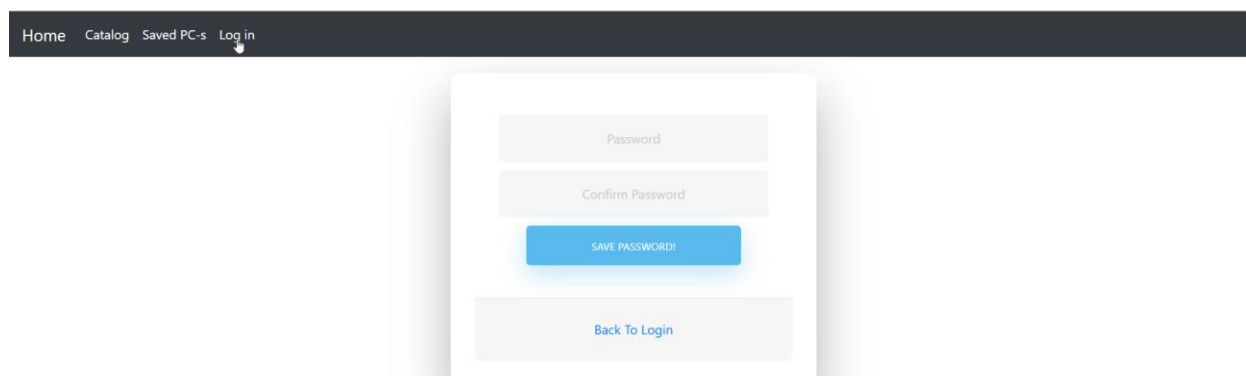




The screenshot shows a web application interface with a dark header bar containing the links 'Home', 'Catalog', 'Saved PC-s', and 'Log in'. Below the header, a white modal form is centered on the page. The form contains a text input field labeled 'Email', a blue button labeled 'SEND EMAIL!', and a link labeled 'Back to login'.

Рисунок 3.6.7 – Забув Пароль?

Вводимо новий пароль та дублюємо його. Натискаємо на кнопку “SAVE PASSWORD”. Якщо все зроблено правильно, то висвітиться зелений надпис з повідомленням. Перейдемо на сторінку авторизації й спробуємо увійти в акаунт.



The screenshot shows the same web application interface as Figure 3.6.7. The 'Log in' link in the header is now highlighted with a mouse cursor. The white modal form is updated to contain two text input fields, 'Password' and 'Confirm Password', a blue button labeled 'SAVE PASSWORD!', and a link labeled 'Back To Login'.

Рисунок 3.6.8 – Зберегти новий пароль

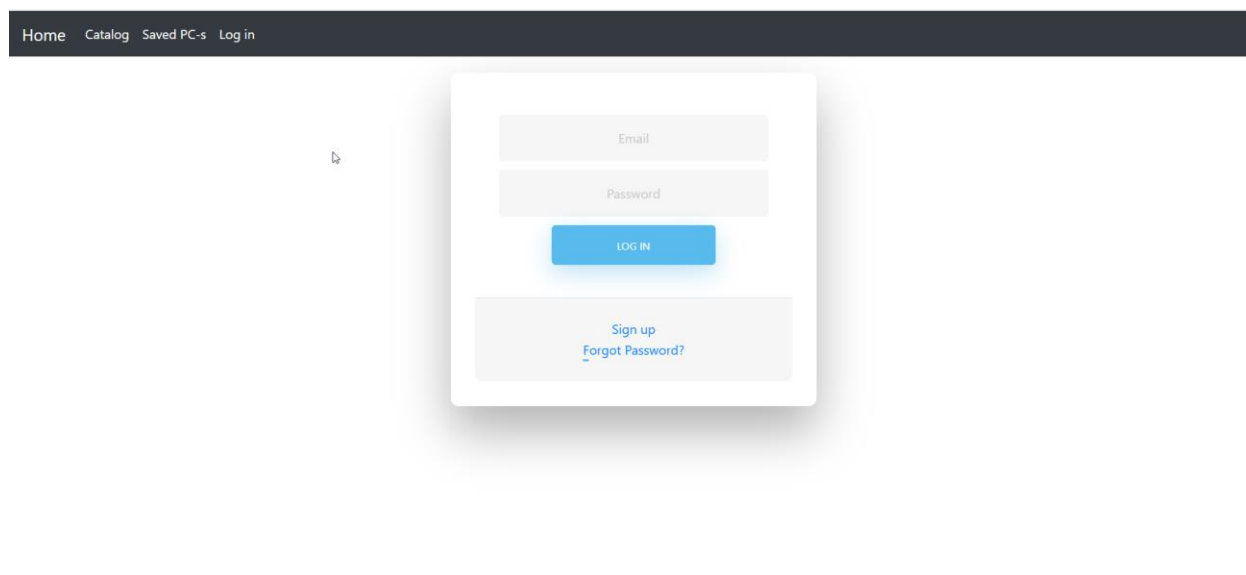
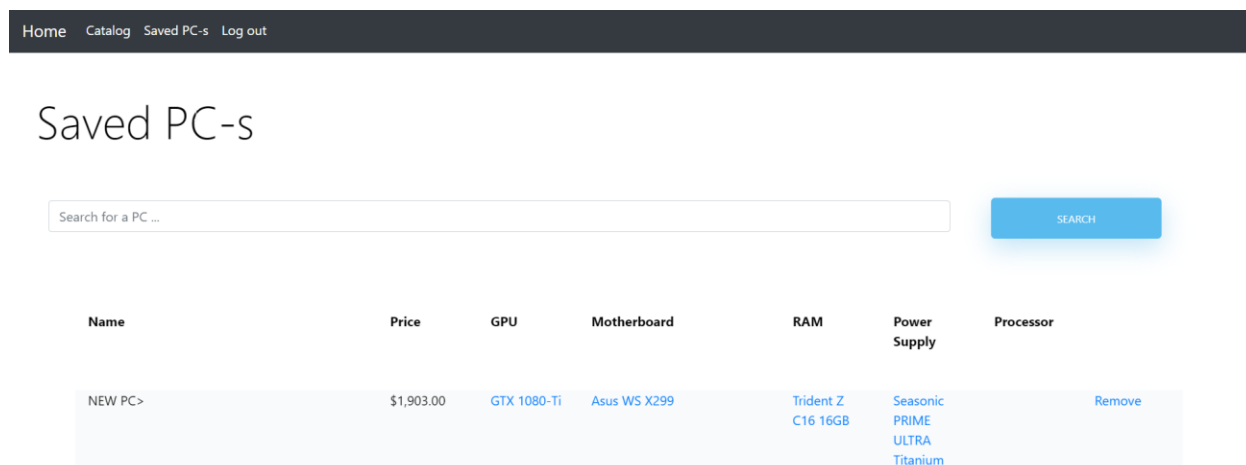


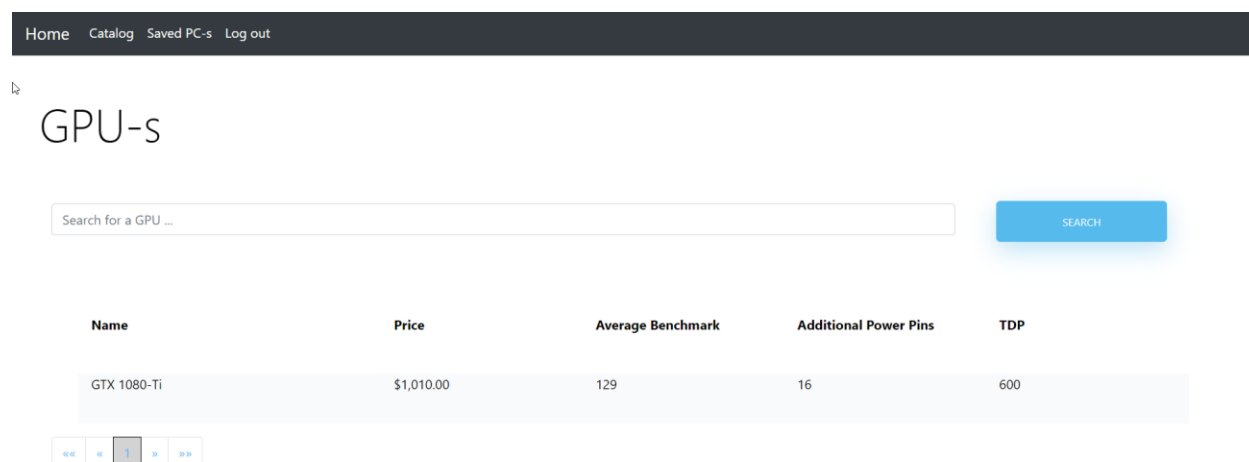
Рисунок 3.6.9 – Авторизація

Зберігаємо збірку кнопкою “Save” та переходимо на сторінку “Saved PC-s”.



### Рисунок 3.6.10 – Збережені комп'ютери

Натискаємо на будь-яку запчастину і нас перенаправляє на сторінку з характеристиками деталі.



### Рисунок 3.6.11 – Характеристики компоненти

Спробуємо перейти в каталог. Тут можна обрати категорії запчастин і побачити всі детальні характеристики.

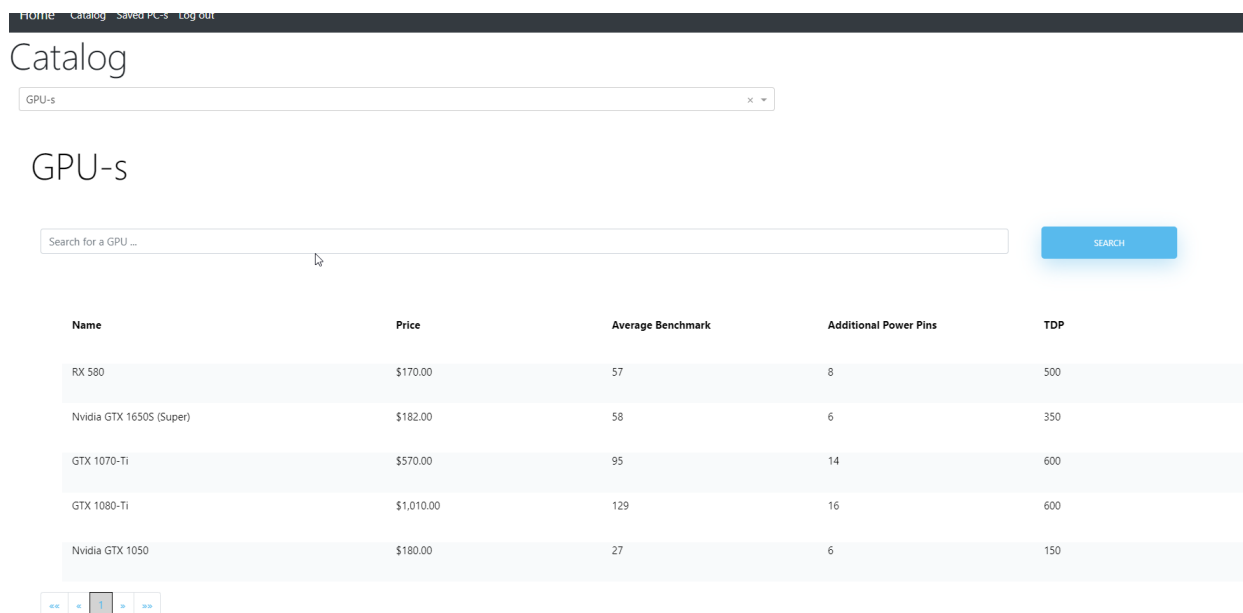


Рисунок 3.6.12 – Каталог

### 3.7 Висновок до розділу 3

У цьому розділі розглянуто розробку головного додатку. Показана реляційна модель та описані усі таблиці. Також є інструкція по використанню вебсайту. Алгоритм, який був використаний для аналізу сумісності переданої збірки був імплементований та протестований.

## Висновки

Для імплементації розробленого додатка була вивчена та засвоєна дуже велика кількість матеріалу. Були проаналізовані всі можливі схожі додатки та виявлені їх недоліки. На основі цих даних був зроблений сервіс.

В результаті використання всіх обраних технологій, зроблений висновок, що вони не даремно знаходяться серед найкращих у світі. Був дуже непоганий досвід DevOps, тому що довелося розгортати всі модулі у хмарі, налаштування великої кількості конфігів.

Також засвоєний досвід зі створення та адміністрування бази даних. Серед великої кількості таблиць довелося робити багато SQL запитів.

В теорії цей сервіс може бути використаний в комерційних цілях, бо аналогів дуже мало і вони не дуже зручні у використанні. Перспективою розвитку є додати інтеграцію third-party сервісів таких як: Rozetka або Amazon для отримання їх актуальної бази даних всіх комплектуючих. Також можна зразу перенаправляти юзерів на оформлення покупки обраної збірки.

По закінченню строків, всі поставлені задачі у технічному завданні були виконані.

Найтяжчим у цьому проєкті було написати алгоритм для основних функцій використовуючи найкращу практику.

## Список використаної літератури

1. Горобей Б. Приклад питань до процесора // Rozetka. 2021. С.1  
[Електронний ресурс]:  
[https://hard.rozetka.com.ua/amd\\_100\\_100000065box/p257442296/questions](https://hard.rozetka.com.ua/amd_100_100000065box/p257442296/questions)
2. Red H. <https://www.redhat.com/en/topics/api/what-is-a-rest-api> // RedHat. 2021.  
С.1 [Електронний ресурс]: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
3. Djuraskovic O. The Growth of The Web // How Many Websites Are There? – The Growth of The Web (1990 – 2021). 2021. С.1 1 [Електронний ресурс]:  
<https://firstsiteguide.com/how-many-websites/#:~:text=There%20are%20currently%20over%201.84,number%20are%20actually%20active%20sites.>
4. PcPartPicker 2021 [Електронний ресурс]: <https://pcpartpicker.com/>
5. Amazon 2021 [Електронний ресурс]: <https://www.amazon.com/>
6. BuildMyPc [Електронний ресурс]: <https://buildmypc.net/>
7. Lenove Custom Laptop [Електронний ресурс]:  
<https://www.lenovo.com/in/en/d/customise-to-order>
8. Таненбаум Э. Архитектура компьютера . Нью-Йорк: Питер, 2015. 816 с.
9. Hong E. How Does Bitcoin Mining Work? // CRYPTOCURRENCY BITCOIN. 2021. С.1 [Електронний ресурс]: <https://www.investopedia.com/tech/how-does-bitcoin-mining->

[work/#:~:text=By%20mining%2C%20you%20can%20earn,are%20added%20to%20the%20blockchain.](#)

10. Beal V. Random Access Memory (RAM) // Computers. 2021. С.1

[Электронный ресурс]: <https://www.webopedia.com/definitions/ram/>

11. AMD 2021 [Электронный ресурс]: <https://www.amd.com>

12. Intel 2021 [Электронный ресурс]: <https://www.intel.com>

13. NVIDIA 2021 [Электронный ресурс]: <https://www.nvidia.com>

14. vidhus138 GTX vs RTX – Which is better? // Geeks For Geeks. 2020. С.1

[Электронный ресурс]: <https://www.geeksforgeeks.org/gtx-vs-rtx-which-is-better/#:~:text=RTX%20stands%20for%20Ray%20Tracing,2018%20and%20uses%20Turing%20architecture.>

15. Brent H. Which RAM Will You Choose for Your PC? // techguided. 2018. С.1

[Электронный ресурс]: <https://techguided.com/how-to-choose-ram/#:~:text=For%20the%20most%20part%2C%20RAM,an%20additional%208GB%20of%20RAM.>

16. Intel What is hyperthreading? // Intel. 2021. С.1 [Электронный ресурс]:

<https://www.intel.ru/content/www/ru/ru/gaming/resources/hyper-threading.html>

17. Power Supply. 2009. С. 1 [Электронный ресурс]:

<https://techterms.com/definition/powersupply>

18. Heroku 2021 [Электронный ресурс]: <https://heroku.com/>

19. Postgresql 2021 [Электронный ресурс]: <https://www.postgresql.org/>

20. Hypertext Transfer Protocol (HTTP) // Protocols. 2021. С.1 [Электронный ресурс]:

[https://www.extrahop.com/resources/protocols/http/#:~:text=What%20is%20HTTP%20\(Hypertext%20Transfer,on%20the%20World%20Wide%20Web.](https://www.extrahop.com/resources/protocols/http/#:~:text=What%20is%20HTTP%20(Hypertext%20Transfer,on%20the%20World%20Wide%20Web.)

21. Mule Soft (2021). What is an API? (Application Programming Interface). Api Strategy, 1. [Электронный ресурс]:

<https://www.mulesoft.com/resources/api/what-is-an-api#:~:text=API%20is%20the%20acronym%20for,you're%20using%20an%20API>

.

22. MVC // TechTerms. 2018. C.1 [Электронный ресурс]:

<https://techterms.com/definition/mvc>

23. What is JSON?. w3schools, C. 1. [Электронный ресурс]:

[https://www.w3schools.com/whatis/whatis\\_json.asp#:~:text=JSON%20stands%20for%20JavaScript%20Object,for%20storing%20and%20transporting%20data](https://www.w3schools.com/whatis/whatis_json.asp#:~:text=JSON%20stands%20for%20JavaScript%20Object,for%20storing%20and%20transporting%20data) 24.

<https://habr.com/ru/company/mobileup/blog/335382/>

25. What is a POJO Class?. baeldung, 1-1. [Электронный ресурс]:

<https://www.baeldung.com/java-pojo-class>

26. SAKOVICH N. 10 Best Web Development Frameworks in 2021 // Sam Solutions. 2021. C.1 [Электронный ресурс]: [https://www.sam-](https://www.sam-solutions.com/blog/web-development-frameworks/)

[solutions.com/blog/web-development-frameworks/](https://www.sam-solutions.com/blog/web-development-frameworks/)

27. Java 2 Platform, Enterprise Edition (J2EE) Overview // Oracle. 2021. C.1

[Электронный ресурс]: <https://www.oracle.com/java/technologies/appmodel.html>

28. JWT 2021 [Электронный ресурс]: <https://jwt.io>