

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ
(ініціали, прізвище)

“ ____ ” червня 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

зі спеціальності

123 «Комп'ютерна інженерія»

на тему: Інформаційна система пошуку електронних компонентів

Виконав (-ла): студент (-ка) IV курсу, групи

КВ-12
(шифр групи)

Гнатюк Софія Валентинівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник ст. викладач каф. СПіСКС Радченко К. О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент ст. викл. каф. обчислювальної техніки ФІОТ Алещенко О.В.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає
запозичень з праць інших авторів без відповідних
посилань.

Студентка _____
(підпис)

Київ – 2025 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма

«Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ

(підпис)

(ініціали, прізвище)

«__» _____ 2024 р.

ЗАВДАННЯ

на дипломний проєкт студентки

Гнатюк Софії Валентинівни

(прізвище, ім'я, по батькові)

1. Тема проєкту Інформаційна система пошуку електронних компонентів _____

керівник проєкту ст. викладач каф. СПіСКС Радченко К. О. _____ ,

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

затверджені наказом по університету від «29» травня 2025 р. №1808-С

2. Термін подання студенткою проєкту _____

3. Вихідні дані до проєкту: див. ТЗ

4. Зміст пояснювальної записки:

- 1) теоретичні відомості для створення інформаційної системи пошуку електронних компонентів;
- 2) опис засобів розробки інформаційної системи;
- 3) розробка інформаційної системи;
- 4) тестування розробки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) :

- 1) загальний алгоритм роботи інформаційної системи;
- 2) схема бази даних;
- 3) алгоритм отримання даних відповідно до запиту;
- 4) схема авторизації та автентифікації;

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, К.Т.Н		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Видача завдання на дипломне проєктування	20.12.2024	
2.	Вивчення літератури за тематикою проєкту	07.01.2025	
3.	Розроблення та узгодження технічного завдання	17.02.2025	
4.	Аналіз існуючих рішень	02.03.2025	
5.	Підготовка матеріалів першого розділу дипломного проєкту	28.03.2025	
6.	Підготовка матеріалів другого та третього розділів дипломного проєкту	20.04.2025	
7.	Підготовка графічної частини дипломного проєкту	25.05.2025	
8.	Оформлення документації дипломного проєкту	03.06.2025	
9.	Попередній огляд матеріалів проєкту на кафедрі	04.06.2025	

Студентка

(підпис)

Софія ГНАТЮК
(ім'я, ПРІЗВИЩЕ)

Керівник проєкту

(підпис)

Костянтин РАДЧЕНКО
(ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (72 с., 49 рис., 2 табл., 3 додатки).

Об'єкт розробки — створення інформаційної системи пошуку електронних компонентів, що дозволяє ефективно організувати електронні компоненти за різноманітними критеріями.

Інформаційна система дозволяє: переглядати та оновлювати каталог електронних компонентів; знаходити об'єкти за допомогою пошуку, фільтрів, сортування тощо; створювати список улюблених електронних компонентів для більш зручного доступу; переглядати актуальні новини зі сфери електроніки із зовнішніх ресурсів. Передбачений механізм подання та обробки скарги у випадку неповної чи некоректної інформації у каталозі. Наявна система авторизації та автентифікації, яка підтримує як звичайних користувачів, так і адміністраторів. В процесі розробки були використані мова програмування Python, фреймворк Django, СУБД PostgreSQL.

В ході розробки:

- проведено аналіз подібних систем для пошуку електронних компонентів, наявних в мережі;
- сформульовані вимоги до інформаційної системи для пошуку електронних компонентів;
- розроблена структура інформаційної системи пошуку електронних компонентів;
- розроблено вебсайт для інформаційної системи для пошуку електронних компонентів;

Використання цієї системи інженерами, виробниками та постачальниками дозволить значно спростити процеси систематизації, пошуку, порівняння електронних компонентів та аналізу їхніх характеристик.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ ЕЛЕКТРОННИХ КОМПОНЕНТІВ, PYTHON, DJANGO, POSTGRESQL, БАЗА ДАНИХ.

ABSTRACT

The qualification work includes an explanatory note (72 pages, 49 images, 2 tables, 3 appendices).

The object of development is the creation of an information system for electronic component search, that provides an effective way of structuring electronic components using various criteria.

The information system allows: viewing and updating the electronic component catalog; finding objects using search, filters, sorting etc; creating a list of favorite electronic components for an easy access; viewing recent news about electronics from external resources. A mechanism for sending and processing reports in case of missing or incorrect data was implemented. Authorization and authentication system handles both ordinary users and administrators. Python programming language, Django framework and PostgreSQL RDBMS were used in the development process.

In the development process:

- an analysis of similar, existing electronic component search systems was carried out;
- requirements were defined for the information system for electronic component search;
- the structure of the information system for electronic component search was developed;
- the website for the information system for electronic component search was developed;

The usage of this system by engineers, manufacturers and distributors may significantly simplify the electronic component structuring, search, comparison processes, as well as characteristics analysis.

Keywords: INFORMATION SYSTEM FOR ELECTRONIC COMPONENT SEARCH, PYTHON, DJANGO, POSTGRESQL, DATABASE.

=Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045440.002 ТЗ	Інформаційна система пошуку електронних компонентів. Технічне завдання	4		
	A4	ІАЛЦ.045440.003 ТП	Інформаційна система пошуку електронних компонентів. Відомість технічного проєкту	2		
	A4	ІАЛЦ.045440.004 ПЗ	Інформаційна система пошуку електронних компонентів. Пояснювальна записка	70		
	A4	ІАЛЦ.045440.005 Д1	Загальна схема алгоритму роботи інформаційної системи. Схема алгоритму	1		
ІАЛЦ.045440.001 ОА						
Змін.	Арк.	№ докум.	Підпис	Дата		
Розробив	Гнатюк С.В.				Літ.	Аркуш Аркушів
Перевірив	Радченко К. О.					1 2
Консуьлт.					КПП ім. Ігоря Сікорського, ФПМ KB-12	
Н. контроль	Клятченко Я.М.					
Зав. каф.	Романкевич В.О.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045440.006 Д2	Схема бази даних.	1		
			Схема структурна			
	A4	ІАЛЦ.045440.007 Д3	Схема алгоритму	1		
			отримання даних			
			відповідно до запиту.			
			Схема алгоритму			
	A4	ІАЛЦ.045440.008 Д4	Схема алгоритму	1		
			авторизації та			
			автентифікації.			
			Схема алгоритму			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал			
Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045440.001 ОА	
					Арк.	2

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....Ошибка! Закладка не определена.
2. ПІДСТАВА ДЛЯ РОЗРОБКИОшибка! Закладка не определена.
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИОшибка! Закладка не определена.
4. ДЖЕРЕЛА РОЗРОБКИ.....Ошибка! Закладка не определена.
5. ТЕХНІЧНІ ВИМОГИ.....Ошибка! Закладка не определена.
6. ЕТАПИ РОЗРОБКИОшибка! Закладка не определена.

					ІАЛЦ.045440.002 ТЗ			
Змін	Арк.	№ докум.	Підпис	Дата	«Інформаційна система пошуку електронних компонентів» <i>Технічне завдання</i>	Літ.	Аркуш	Аркушів
Розробив		Гнатюк С.В.						
Перевірів		Радченко К. О.					1	4
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ КВ-12		
Затвердив		Романкевич В.О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Найменування роботи: «Інформаційна система пошуку електронних компонентів».

Галузь застосування: інженерія, систематизація електронних компонентів, комп'ютерна електроніка.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою роботи є створення інформаційної системи, що систематизує дані про електронні компоненти та дозволяє користувачам шукати, порівнювати та аналізувати характеристики електронних компонентів у зручний спосіб.

4. ДЖЕРЕЛА РОЗРОБКИ

Інформація взята з технічної та науково-технічної літератури, технічної документації, публікацій у періодичних виданнях та електронних статей в Інтернеті.

					ІАЛЦ.045440.002 ТЗ			
Змін	Арк.	№ докум.	Підпис	Дата	«Інформаційна система пошуку електронних компонентів» Технічне завдання	Літ.	Аркуш	Аркушів
Розробив	Гнатюк С.В.						1	4
Перевірив	Радченко К. О.							
Н. контроль	Клятченко Я.М.							
Затвердив	Романкевич В.О.							
						КПІ ім. Ігоря Сікорського, ФПМ КВ-12		

5. ТЕХНІЧНІ ВИМОГИ

1. Вимоги до програмного продукту, що розробляється

- можливість переглядати, додавати, оновлювати та видаляти електронні компоненти та їхні категорії;
- зручний пошук на основі ключових слів, фільтрів та сортування;
- система авторизації та автентифікації;
- можливість модерації контенту та механізм подання скарг;
- зрозумілий інтерфейс, зручний у використанні.

2. Вимоги до апаратного забезпечення

- наявність доступу до мережі Інтернет;
- оперативна пам'ять обсягом 1 Гб або більше.

1. Вимоги до програмного та апаратного забезпечення користувача

- один із сучасних веббраузерів.

					ІАЛЦ.045440.002 ТЗ			
Змін	Арк.	№ докум.	Підпис	Дата	«Інформаційна система пошуку електронних компонентів» <i>Технічне завдання</i>	Літ.	Аркуш	Аркушів
Розробив		Гнатюк С.В.					1	4
Перевірив		Радченко К. О.						
Н. контроль		Клятченко Я.М.						
Затвердив		Романкевич В.О.				КПІ ім. Ігоря Сікорського, ФПМ КВ-12		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Пошук та ознайомлення з літературою	07.11.2024
2.	Розробка та узгодження технічного завдання	17.12.2024
3.	Пошук та аналіз існуючих рішень	02.02.2025
4.	Підготовка матеріалів до першого розділу дипломного проєкту	28.03.2025
5.	Підготовка матеріалів до другого розділу дипломного проєкту	20.04.2025
6.	Підготовка графічної частини дипломного проєкту	25.05.2025
7.	Оформлення звіту та документації дипломного проєкту	03.06.2025
8.	Попередній огляд матеріалів дипломної роботи на кафедрі	04.06.2025

					ІАЛЦ.045440.002 ТЗ			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив		Гнатюк С.В.			«Інформаційна система пошуку електронних компонентів» Технічне завдання	Літ.	Аркуш	Аркушів
Перевірив		Радченко К. О.					1	4
						КПІ ім. Ігоря Сікорського, ФПМ КВ-12		
Н. контроль		Клятченко Я.М.						
Затвердив		Романкевич В.О.						

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045440.004 ПЗ	Інформаційна система для пошуку електронних компонентів. Пояснювальна записка	70		
	A4	ІАЛЦ.045440.005 Д1	Загальна схема алгоритму роботи інформаційної системи. Схема алгоритму	1		
	A4	ІАЛЦ.045440.006 Д2	Схема бази даних. Схема структурна	1		
	A4	ІАЛЦ.045440.007 Д3	Схема алгоритму отримання даних відповідно до запиту. Схема алгоритму	1		
	A4	ІАЛЦ.045440.008 Д4	Схема алгоритму авторизації та автентифікації. Схема алгоритму	1		

					ІАЛЦ.045440.003 ТП									
Змін.	Арк.	№ докум.	Підпис	Дата	Інформаційна система пошуку електронних компонентів Відомість технічного проєкту					Літ.		Аркуш	Аркушів	
Розробив	Гнатюк С.В.												1	2
Перевірив	Радченко К. О.													
Консульт.														
Н. контроль	Клятченко Я.М.													
Зав. каф.	Романкевич В.О.									КПІ ім. Ігоря Сікорського, ФПМ KB-12				

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Інформаційна система пошуку електронних компонентів

Київ – 2025

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	4
ВСТУП	5
1. ТЕОРЕТИЧНІ ВІДОМОСТІ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ ЕЛЕКТРОННИХ КОМПОНЕНТІВ	6
1.1. Огляд сфери розробки	6
1.2. Теоретичні відомості про розробку інформаційних систем.....	7
1.3. Дослідження доступних аналогів	11
1.4. Постановка задачі	23
Висновки до розділу 1	25
2. ОПИС ЗАСОБІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
2.1. Вибір мови програмування	27
2.2. Вибір фреймворку	30
2.3. Вибір СУБД	33
Висновки до розділу 2	37
3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	39
3.1. Створення моделі «сутність-зв’язок»	39
3.2. Проєктування бази даних	42
3.3. Визначення структури проєкту Django.....	47
3.4. Розробка інформаційної системи	48
3.5. Подальші плани щодо розвитку продукту	55
Висновки до розділу 3	56
4. ТЕСТУВАННЯ РОЗРОБКИ.....	57
4.1. Тестування основного функціоналу.....	57
4.2. Тестування додаткового функціоналу	61
4.3. Тестування новинного функціоналу	64
Висновки до розділу 4	65

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API – Application Programming Interface (прикладний програмний інтерфейс) – спосіб спілкування комп'ютерних програм між собою.

HTTP – Hypertext Transfer Protocol (гіпертекстовий протокол передачі) – протокол прикладного рівня моделі OSI.

Модель OSI (Open System Interconnection, взаємодія відкритих систем) – абстрактна модель, що представляє поділ мереж на рівні.

СУБД – система управління базою даних.

БД – база даних.

SQL – Structured Query Language (структурована мова запитів) – мова для взаємодії із базою даних.

REST – Representational State Transfer (передача репрезентативного стану) – підхід до архітектури прикладних програм для отримання доступу до інформаційного ресурсу.

HTML – HyperText Markup Language (мова розмітки гіпертексту) – стандартизована мова для розмітки документів, що представляють вебсторінки у браузері.

						4

ВСТУП

У сучасному світі, де технології розвиваються з небаченою швидкістю, а обсяги доступної інформації постійно зростають, перед суспільством та спеціалістами постає не лише завдання збору даних, але й надзвичайно важливе питання їх ефективної організації. У таких умовах стає критично необхідним не просто накопичувати інформацію, а вміти її грамотно структурувати, оперативно обробляти та подавати у зрозумілому, зручному і доступному для кінцевого користувача вигляді. Ця проблема є особливо гострою у технічних і високотехнологічних галузях, де точність, швидкість обробки й легкий доступ до даних відіграють ключову роль у прийнятті рішень, проектуванні, розробці та впровадженні інженерних рішень.

Відповідно, постає потреба у створенні спеціалізованих інструментів, здатних забезпечити впорядкування великої кількості технічної інформації, зробити її більш керованою, стандартизованою та доступною для аналітики. Такі інструменти стають не просто корисними додатками, а фундаментально важливою складовою ефективної роботи сучасного інженера, технічного фахівця чи дослідника.

Метою даної кваліфікаційної роботи є розробка інформаційної системи для пошуку електронних компонентів. Така система покликана не лише впорядкувати різноманітні відомості про електронні компоненти, але й забезпечити зручний інтерфейс для швидкого пошуку, перегляду та аналізу даних. У результаті вона має стати ефективним інструментом для інженерів, студентів технічних спеціальностей, розробників та інших фахівців, які працюють із електронікою, допомагаючи їм швидко знаходити необхідні компоненти, порівнювати їх характеристики та приймати обґрунтовані технічні рішення.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ ЕЛЕКТРОННИХ КОМПОНЕНТІВ

1.1. Огляд сфери розробки

У сучасному світі, де інформація відіграє ключову роль, впорядкування та швидкий доступ до даних є необхідністю в багатьох сферах. Це стосується і електроніки, де широкий вибір компонентів з різними характеристиками створює потребу в ефективних інструментах для їхнього пошуку та аналізу.

Систематизація електронних компонентів є важливим завданням для інженерів, виробників і постачальників, оскільки дозволяє швидко знаходити потрібні деталі, аналізувати їхні характеристики та здійснювати закупівлі.

Електронні компоненти мають велику кількість параметрів, які визначають їхню функціональність, сумісність і сферу застосування. До них належать електричні характеристики (наприклад, напруга, струм, опір, ємність), фізичні властивості (розміри, форма корпусу, тип монтажу) та експлуатаційні особливості (робочий температурний діапазон, стійкість до зовнішніх впливів).

Окрім того, компоненти поділяються на різні категорії, такі як пасивні (резистори, конденсатори, індуктивні компоненти), активні (транзистори, мікросхеми, діоди), сенсори, з'єднувачі тощо [7]. Велика різноманітність та складність цих елементів створює додаткові виклики під час розробки ефективного онлайн-каталогу, який повинен забезпечувати зручну навігацію, правильну класифікацію та можливість швидкого пошуку за параметрами.

Для впорядкування та зберігання таких даних використовуються інформаційні системи у вигляді онлайн-каталогів. Це вебресурси, що містять структуровану інформацію та дозволяють користувачам знаходити потрібні компоненти за допомогою пошуку, фільтрів або категоризації. Онлайн-каталоги застосовуються в різних сферах – від електронної комерції до наукових баз даних і довідкових систем. Наприклад, каталоги електронних

магазинів містять детальні характеристики товарів, зображення, ціни та відгуки, тоді як у наукових і освітніх системах вони використовуються для впорядкування літературних джерел, дослідницьких матеріалів і архівів.

Таким чином, впорядкування даних про електронні компоненти за допомогою онлайн-каталогів значно спрощує процес проєктування, виробництва та закупівлі, роблячи роботу фахівців більш ефективною та зручною.

1.2. Теоретичні відомості про розробку інформаційних систем

Онлайн-каталог електронних компонентів є прикладом інформаційної системи.

Інформаційна система – це інтегрований набір компонентів, призначених для збору, зберігання, обробки та надання даних і інформації. Вона підтримує операційну діяльність, управління та процес ухвалення рішень в організаціях. Основними компонентами є апаратне та програмне забезпечення, телекомунікації, бази даних, сховища даних, людські ресурси та процедури.

Інформаційні системи відіграють ключову роль у керуванні бізнес-процесами, взаємодії з клієнтами та постачальниками, а також у забезпеченні конкурентоспроможності на ринку. Вони сприяють виконанню різних функцій, таких як обробка транзакцій, підтримка ухвалення рішень і бізнес-аналітика. Такі системи є необхідними як для підвищення ефективності організації, так і для стратегічного планування, оскільки дозволяють аналізувати дані, прогнозувати тенденції та приймати обґрунтовані рішення.

Інформаційні системи значно еволюціонували завдяки технологічному прогресу. Спочатку механічні системи, такі як табулятор для перепису населення Германа Голлеріта, автоматизували обробку даних. Поява комп'ютерів, зокрема UNIVAC I у 1950-х роках, ознаменувала перехід до електронної обробки даних. У 1970-х роках персональні комп'ютери зробили інформаційні системи доступними для малого бізнесу та індивідуальних

						7

користувачів.

Інтернет, що з'явився наприкінці ХХ століття, здійснив революцію доступу до інформації та комунікації, відкривши можливості для електронної комерції та цифрової взаємодії. Сьогодні мобільні пристрої та бездротові мережі ще більше розширюють охоплення інформаційних систем, підтримуючи широкий спектр діяльності та змінюючи способи соціальної взаємодії та бізнес-моделі.

Бізнес-фірми та інші організації покладаються на інформаційні системи для виконання та управління своїми операціями, взаємодії з клієнтами та постачальниками, а також для конкуренції на ринку. Інформаційні системи використовуються для управління міжорганізаційними ланцюгами постачання та електронними ринками. Наприклад, корпорації застосовують інформаційні системи для обробки фінансових рахунків, управління людськими ресурсами та для досягнення потенційних клієнтів через онлайн-промоції. Багато великих компаній побудовані виключно навколо інформаційних систем. Серед них eBay - аукціонний ринок; Amazon - електронний торговий центр і постачальник хмарних обчислювальних послуг; і Google - компанія-власник пошукової системи, більшість доходів якої надходить від контекстної реклами за ключовими словами в Інтернет-пошуках.

Уряди використовують інформаційні системи для надання зручних та ефективних послуг громадянам. Цифрові товари, такі як електронні книги, відеопродукція та програмне забезпечення, а також інтерактивні продукти, як-от ігри та соціальні мережі, надаються за допомогою інформаційних систем. Люди також залежать від інформаційних систем, зазвичай на основі Інтернету, для ведення значної частини свого особистого життя: для соціалізації, навчання, покупок, банківських операцій та розваг.

З огляду на значну роль інформаційних систем у сучасному цифровому середовищі, особливе місце займають вебтехнології, які забезпечують доступ до інформації та взаємодію користувачів із різними сервісами. Одним із

						8

ключових елементів цього середовища є вебсайти.

Вебсайт – це структурована сукупність вебсторінок, які пов’язані між собою гіперпосиланнями та доступні через мережу Інтернет за допомогою веббраузера. Вебсайти можуть містити текстову, графічну, аудіо- та відеоінформацію, а також інтерактивні елементи, що забезпечують взаємодію з користувачами. З технічної точки зору, вебсайт складається з фронтенд-частини (інтерфейсу користувача) та бекенд-частини (серверної логіки, баз даних та інших компонентів, що забезпечують його функціонування).

Вебсайти виконують широкий спектр функцій, зокрема надання інформації, електронну комерцію, управління бізнес-процесами, соціальну взаємодію та розважальні сервіси. Вони є невід’ємною частиною сучасних інформаційних систем, оскільки дозволяють забезпечувати ефективну комунікацію, автоматизацію операцій та доступ до цифрових ресурсів. Загалом, вебсайти можна поділити на кілька категорій відповідно до їхнього призначення та функціоналу:

- 1) інформаційні ресурси – вебсайти, що створені для надання користувачам доступу до інформації у зручному, структурованому вигляді. До цієї категорії належать новинні, освітні, енциклопедичні та інші ресурси;
- 2) корпоративні вебсайти – вебсайти, що призначені для представлення комерційних компаній, громадських організацій чи державних установ;
- 3) платформи для електронної комерції – вебсайти, що дозволяють користувачам здійснювати онлайн-покупки товарів або послуг та здійснювати оплату через електронні платіжні системи;
- 4) соціальні мережі – вебсайти, створені для спілкування користувачів та обміну інформацією;
- 5) мультимедійні та розважальні платформи – вебсайти, призначені для публікації та перегляду мультимедійного контенту (фільмів, відео, музики тощо);

- 6) вебдодатки – інтерактивні сайти, що виконують функції програмного забезпечення, яке запускається через браузер користувача[1].

Онлайн-каталоги найчастіше належать до інформаційних ресурсів, проте їх також використовують у сфері електронної комерції та професійних систем управління даними. Для ефективного функціонування таких платформ застосовуються бази даних, що забезпечують структуроване зберігання інформації та оптимізацію запитів користувачів.

Для забезпечення ефективної роботи вебсайтів використовується клієнт-серверна архітектура, яка є основною моделлю взаємодії в сучасних інформаційних системах.

Клієнт-серверна архітектура – це модель обчислень, у якій клієнт (користувацький пристрій або програма) ініціює запити до сервера, який обробляє ці запити та надає необхідні ресурси чи послуги. У контексті вебтехнологій клієнтом зазвичай виступає веббраузер або мобільний додаток, а сервером – віддалений комп'ютер або кластер серверів, що зберігають дані, виконують обчислення та забезпечують відповідний контент.

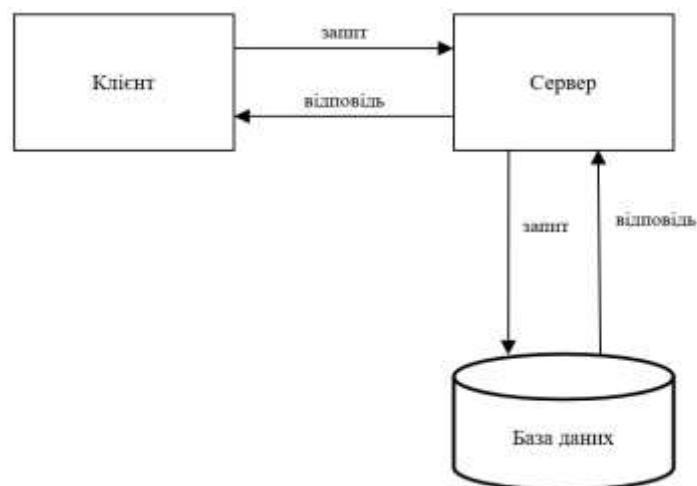


Рис.1.1 – Схема роботи системи із клієнт-серверною архітектурою

Клієнт-серверна модель дозволяє розподіляти навантаження між пристроями та централізовано керувати обробкою даних, що сприяє

масштабованості, безпеці та ефективному управлінню ресурсами. Вона є основою роботи вебдодатків, хмарних сервісів, електронної комерції та інших цифрових платформ, які інтегруються в сучасні інформаційні системи.

Оскільки клієнт-серверна архітектура дозволяє організовувати взаємодію між користувачами та вебсервісами, вебсайти стали основним засобом отримання та обміну інформацією. Люди використовують їх для пошуку знань, спілкування, здійснення покупок, роботи з документами та багатьох інших завдань. Для підвищення зручності та ефективності пошуку інформації важливо, щоб вона була структурованою та впорядкованою. Чітка організація даних дозволяє користувачам швидко знаходити потрібні відомості, а системам – ефективно обробляти запити. Одним із ключових механізмів, що допомагає в цьому, є використання спеціалізованих засобів для систематизації та зберігання інформації, зокрема баз даних. Вони дозволяють організовувати великі обсяги даних у зручній формі, забезпечуючи швидкий доступ і коректне керування інформацією.

1.3. Дослідження доступних аналогів

Перш ніж розпочати проєктування інформаційної системи-каталогу електронних компонентів, необхідно розглянути вже наявні аналоги та провести дослідження їхнього функціоналу. Зокрема, до наявних аналогів належать такі вебсайти, як “Otopart”, “Mouser Electronics”, “DigiKey” тощо.

Вебсайт “Otopart” представляється як «найбільш точна та зрозуміла пошукова система для даних про електронні компоненти» [3] (рис. 1.2).



Рис. 1.2 – Головна сторінка вебсайту “Otopart”

На головній сторінці сайту можна знайти можливість пошуку компонентів за категорією, виробником або за дистриб'ютором. Також можна переглянути список найбільш популярних компонентів (рис. 1.3).

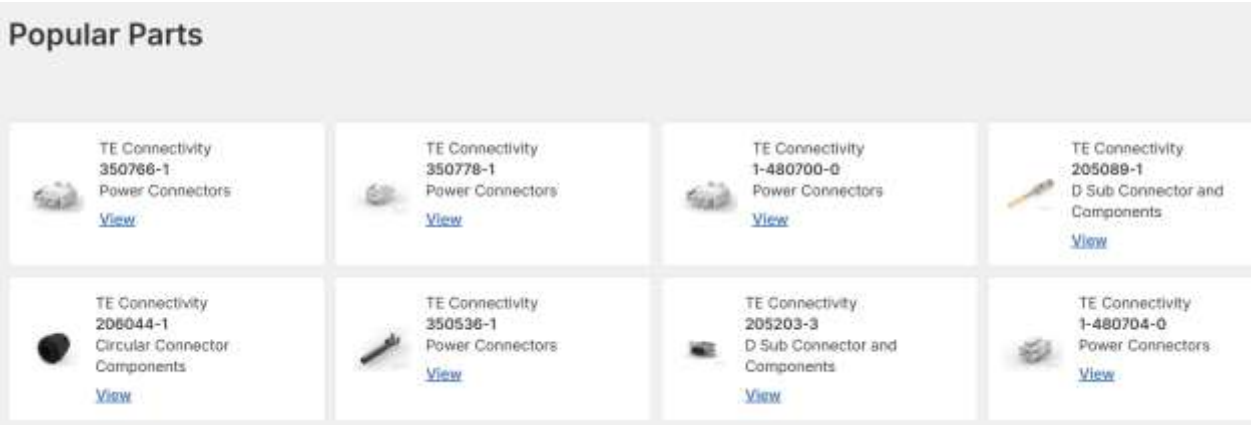


Рис. 1.3 – Перелік популярних компонентів на сайті “Ostorart”

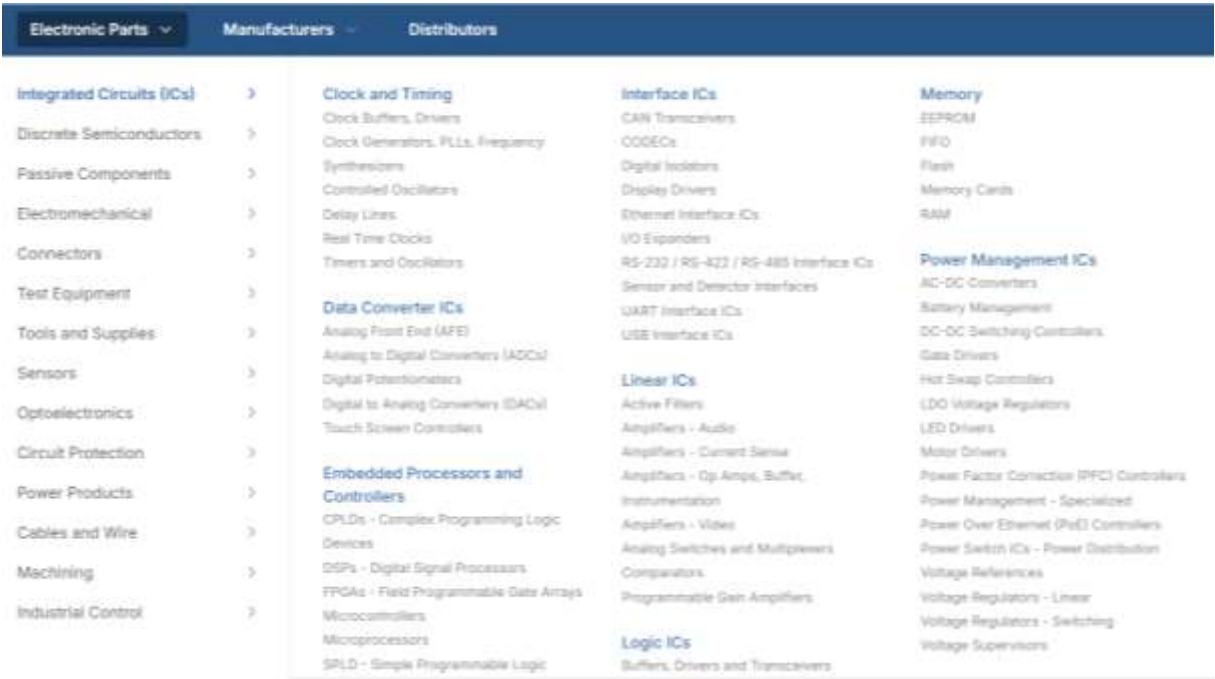


Рис. 1.4 – Меню категорій на сайті “Ostorart”

Є можливість переглянути топ-50 виробників електронних компонентів, а також перейти до повного списку виробників (рис. 1.5).

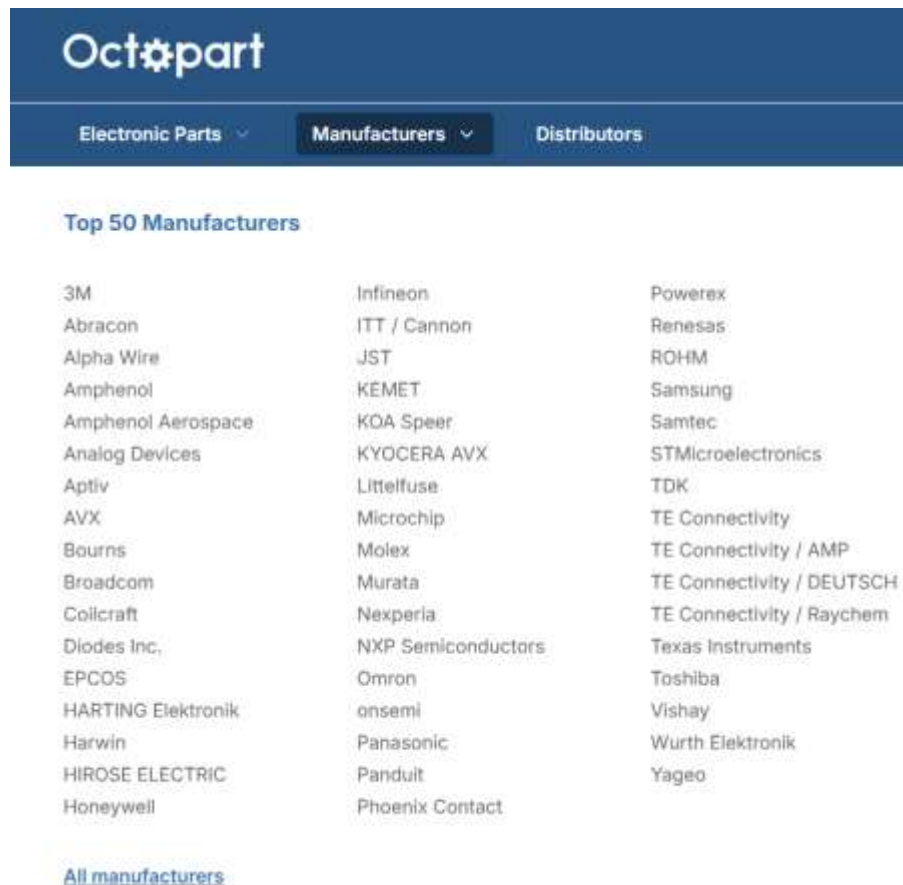


Рис. 1.5 – Меню виробників на вебсайті “Octopart”

Після вибору конкретної категорії користувач переходить на сайті “Octopart” на сторінку із коротким описом цієї категорії, відповідними топ-виробниками, та переліком усіх компонентів даної категорії (рис. 1.6).

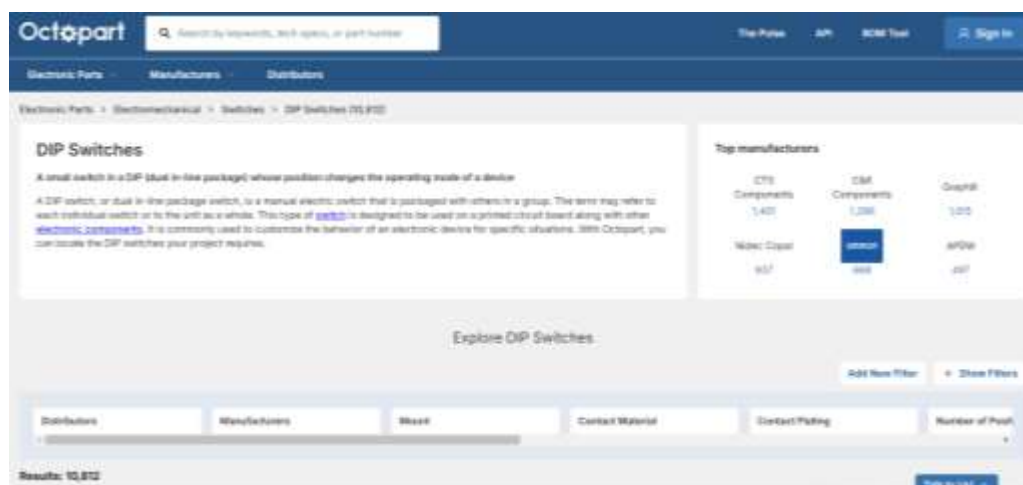
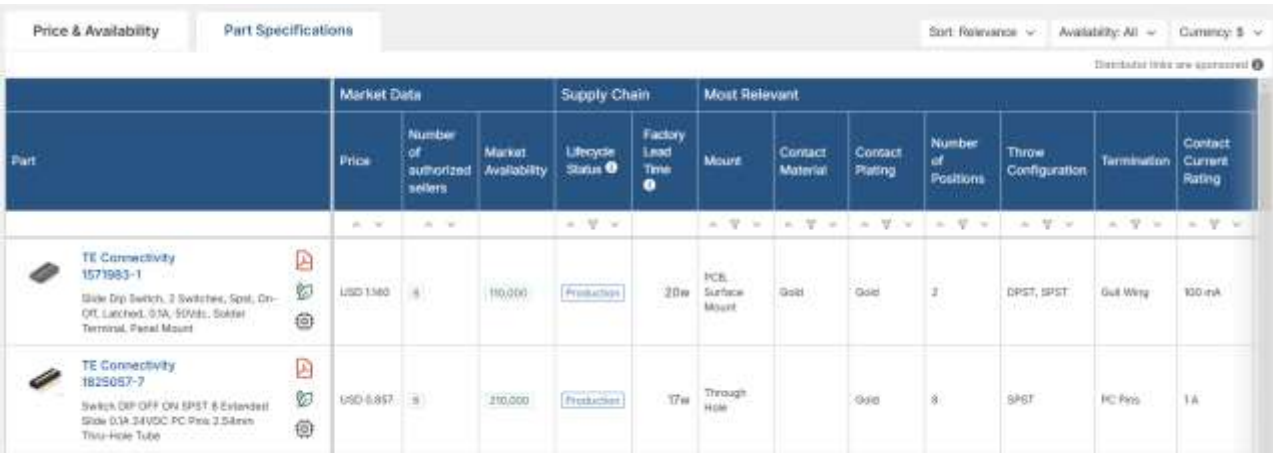


Рис. 1.6 – Сторінка підкатегорії “Electromechanical/Switches/DIP Switches”

Існує два варіанти перегляду списку електронних компонентів: «Ціна і доступність» та «Характеристики компонентів». Залежно від побажань користувача, можна переглянути ціни на подібні компоненти в різних продавців, або переглянути список компонентів, відсортованих чи відфільтрованих за певними характеристиками (рис. 1.7).



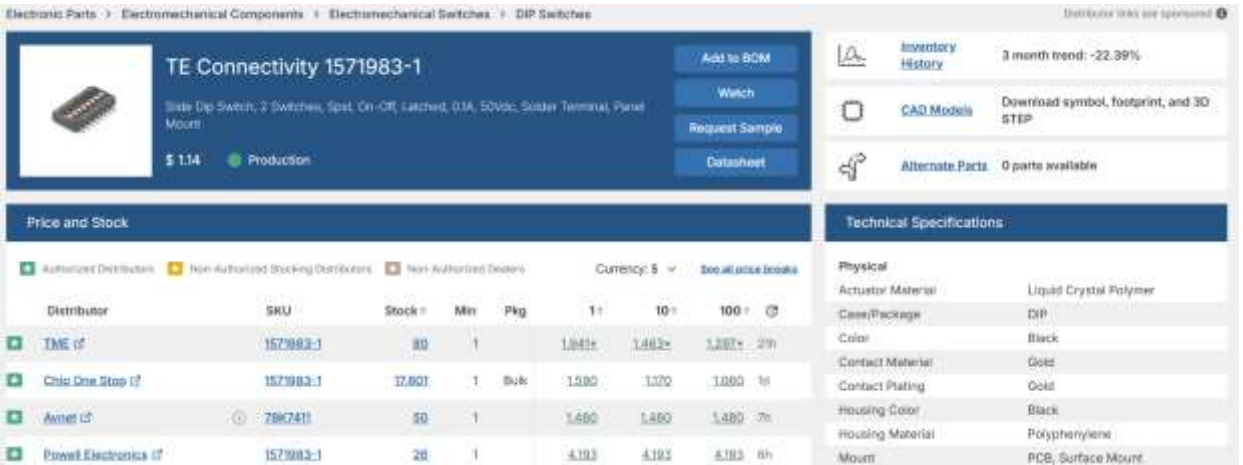
The screenshot shows the Octopart website interface. At the top, there are tabs for 'Price & Availability' and 'Part Specifications'. Below these are filters for 'Sort: Relevance', 'Availability: All', and 'Currency: \$'. The main table lists components with columns for Part, Price, Number of authorized sellers, Market Availability, Lifecycle Status, Factory Lead Time, Mount, Contact Material, Contact Plating, Number of Positions, Throw Configuration, Termination, and Contact Current Rating. Two components are visible: TE Connectivity 1571983-1 and TE Connectivity 1825057-7.

Part	Price	Number of authorized sellers	Market Availability	Lifecycle Status	Factory Lead Time	Mount	Contact Material	Contact Plating	Number of Positions	Throw Configuration	Termination	Contact Current Rating
TE Connectivity 1571983-1 Slide Dip Switch, 2 Switches, Spst, On-Off, Latched, 0.1A, 50Vdc, Solder Terminal, Panel Mount	USD 1.560	8	110,000	Production	28w	PCB, Surface Mount	Gold	Gold	2	DPST, SPST	Gull Wing	100 mA
TE Connectivity 1825057-7 Switch DIP OFF ON SPST 6 Extended Slide 0.1A 24VDC PC Pins 2.54mm Thru-Hole Tube	USD 6.857	8	210,000	Production	17w	Through Hole		Gold	6	SPST	PC Pins	1 A

Рис. 1.7 – Огляд списку компонентів за характеристиками на сайті “Octopart”

Даний ресурс не є інтернет-магазином, але надає інформацію про ціни та наявність електронних компонентів на платформах інтернет-комерції[13].

При переході на сторінку конкретного компонента, користувач може побачити зображення компонента, опис, наявність та ціни на інших платформах, перелік подібних елементів тощо (рис. 1.8).



The screenshot shows the product page for TE Connectivity 1571983-1. It includes a product image, description, price (\$1.14), and production status. There are buttons for 'Add to BOM', 'Watch', 'Request Sample', and 'Datasheet'. The 'Price and Stock' section shows a table of distributors and their prices. The 'Technical Specifications' section lists physical and electrical properties.

Distributor	SKU	Stock	Min	Pkg	1	10	100	250
TME cz	1571983-1	80	1		1.841z	1.463z	1.187z	27h
Chia One Stop cz	1571983-1	17,601	1	8x8	1.980	1.370	1.080	1h
Amnet cz	7987411	50	1		1.680	1.680	1.680	7h
Power Electronics cz	1571983-1	26	1		4.193	4.193	4.193	6h

Technical Specifications	
Physical	
Actuator Material	Liquid Crystal Polymer
Case/Package	DIP
Color	Black
Contact Material	Gold
Contact Plating	Gold
Housing Color	Black
Housing Material	Polyphenylene
Mount	PCB, Surface Mount

Рис. 1.8 – Сторінка електронного компонента на сайті “Octopart”

Іншим подібним ресурсом є вебсайт “Mouser Electronics”. На відміну від попередньої платформи, “Mouser Electronics” також є дистриб’ютором, а не лише каталогом електронних компонентів[11]. На головній сторінці можна побачити меню товарів та виробників, переглянути список найновіших надходжень, перелік популярних категорій та пошук товарів за номером або за ключовими словами (рис. 1. 9).



Рис. 1.9 – Головна сторінка вебсайту “Mouser Electronics”

Якщо обрати певну категорію, на сайті “Mouser Electronics” можна потрапити на сторінку цієї категорії та побачити короткий опис категорії, перелік підкатегорій та перелік товарів (рис. 1.10)

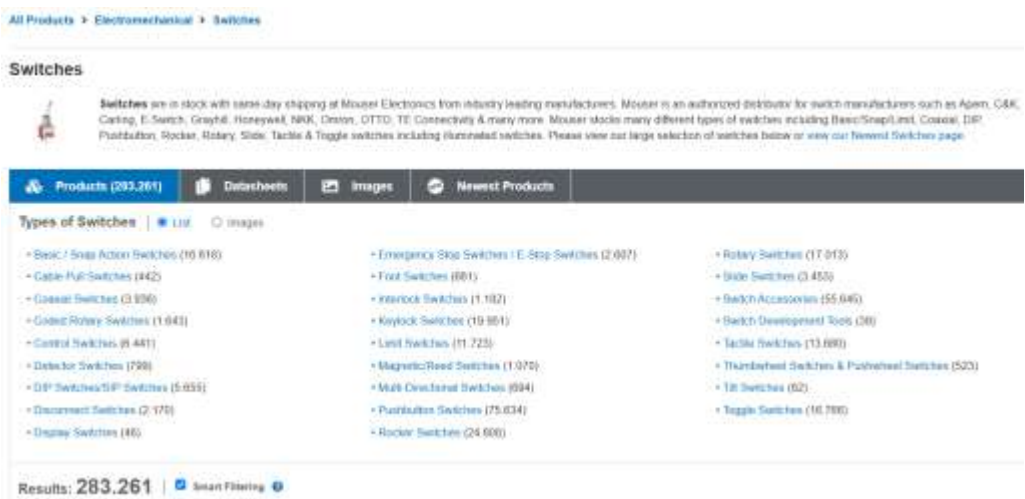


Рис. 1.10 – Сторінка категорії “Electromechanical/Switches”

Нижче на цій сторінці можна побачити перелік відповідних товарів, до яких можна застосувати фільтри або сортування за різними параметрами, а також шукати електронні компоненти серед результатів (рис. 1.11).

The screenshot shows a search results page on the Mouser Electronics website. At the top, there is a search bar with the text 'Search within results' and a filter section with checkboxes for 'In Stock', 'Normally Stocked', 'Active', 'New Products', and 'RoHS Compliant'. Below this is a table of search results. The table has columns for 'Select', 'Image', 'Part #', 'Mfr.', 'Description', 'Datasheet', 'Availability', 'Pricing (EUR)', 'Qty.', 'RoHS', 'Product Type', 'Type', and 'Contact Form'. Two products are visible: 'MT- Part No. B3FS-1066P' and 'BS- Part No. O2EW-SC02MS'. Each product row includes a checkbox, an image, the part number, manufacturer (Omron Electronics), description, a link to the datasheet, availability status, pricing table, quantity input field, a 'Buy' button, a 'Details' link, and the product type and contact form link.

Select	Image	Part #	Mfr.	Description	Datasheet	Availability	Pricing (EUR)	Qty.	RoHS	Product Type	Type	Contact Form
☐		MT- Part No. B3FS-1066P Mouser Part No. B3FS-1066P New Product	Omron Electronics	Tactile Switches Tactile Switch with integrated key top, 2.55N	Datasheet	In Stock	Qty Type 1 1.80 € 10 1.40 € 100 1.30 € 250 1.10 € 500 1.00 €	Min: 1 Mult: 1	Buy	Details	Tactile Switches	SPST
☐		BS- Part No. O2EW-SC02MS Mouser Part No. O2EW-SC02MS New Product	Omron Electronics	Basic / Snap Action Switches Sealed Ultra Subminiature Back Switch UL/UL approved versions Pin plunger/Molded lead wires downwards/SPST-NC/MS-screw mounting models	Datasheet	In Stock	1 8.25 € 10 6.80 € 25 4.75 € 100 4.30 € 250 4.00 €	Min: 1 Mult: 1	Buy	Details	Snap Action Switches	SPST-NC

Рис.1.11 – Перелік товарів певної категорії на сайті “Mouser Electronics”

Також є можливість обрати кілька товарів та переглянути порівняльну таблицю (рис.1.12). Таблиця містить інформацію про характеристики товару, ціни та наявність на складі.

Compare Products		
Product Info:		
Image		
Select ☐	☐	☐
Mouser Part No.	B3FS-1066P	B3S-02EW-SC02MS
Mfr. % Part No.	B3FS-1066P	O2EW-SC02MS
Manufacturer	Omron Electronics	Omron Electronics
Description	Tactile Switches Tactile Switch with integrated key top, 2.55N	Basic / Snap Action Switches Sealed Ultra Subminiature Back Switch UL/UL approved versions Pin plunger/Molded lead wires downwards/SPST-NC/MS-screw mounting models
Lifecycle	New Product	New Product
Datasheet	B3FS-1066P Datasheet (PDF)	O2EW-SC02MS Datasheet (PDF)
RoHS	RoHS Details	RoHS Details
Specifications		
Actuator	Square	Plunger Pin
Brand	Omron Electronics	Omron Electronics
Color	Black	-
Contact Form	SPST	SPST-NC
Contact Plating	Silver	-

Рис. 1.12 – Порівняльна таблиця двох електронних компонентів на сайті “Mouser Electronics”

Сторінка з описом електронного компонента надає дані про характеристики, ціни, наявність на складі та перелік рекомендованих товарів (рис. 1.13).

The screenshot shows the product page for the Omron B3FS-1095P tactile switch on the Mouser Electronics website. The page is divided into several sections:

- Product Information:** Includes the Mouser No. (653-B3FS-1095P), Mfr. No. (B3FS-1095P), Mfr. (Omron Electronics), and a description: "Tactile Switches Tactile Switch with integrated key top, 2.55N".
- Stock and Lead Time:** Shows "In Stock: 883" and "Factory Lead Time: 7 Weeks".
- Pricing (EUR):** A table showing unit prices and estimated prices for various quantities.

Qty.	Unit Price	Est. Price
1	1.89 €	1.89 €
10	1.48 €	14.80 €
100	1.20 €	120.00 €
250	1.10 €	275.00 €
500	1.05 €	525.00 €
1,000	0.968 €	968.00 €
2,500	0.976 €	2,440.00 €

Рис. 1.13 – Сторінка окремого товару на сайті “Mouser Electronics”

Також даний сайт має типовий функціонал інтернет-магазину: кошик покупця, замовлення товару тощо.

Наступний подібний ресурс – “DigiKey”. Це вебсайт дистриб’ютора електронних компонентів[5]. На головній сторінці сайту знаходиться скорочений перелік категорій (та посилання на повний перелік), меню виробників, перелік обраних виробників тощо (рис. 1. 14).

The screenshot shows the homepage of the DigiKey website. The page is divided into several sections:

- Navigation:** Includes links for Products, Manufacturers, Resources, and Request a Quote.
- Products:** A list of product categories such as Automotive & Control, Audio, Communications, and more.
- Central Banner:** A promotional banner for Raspberry Pi with the text "Everyone loves Pi" and a "Shop now" button.
- Tools, Services, and Content:** Sections for various tools, services, and content related to electronics.

Рис. 1.14 – Головна сторінка вебсайту “DigiKey”

Нижче на цій сторінці можна побачити перелік спеціальних інструментів, сервісів, тощо доступних на сайті та перелік обраних товарів (рис. 1.15).

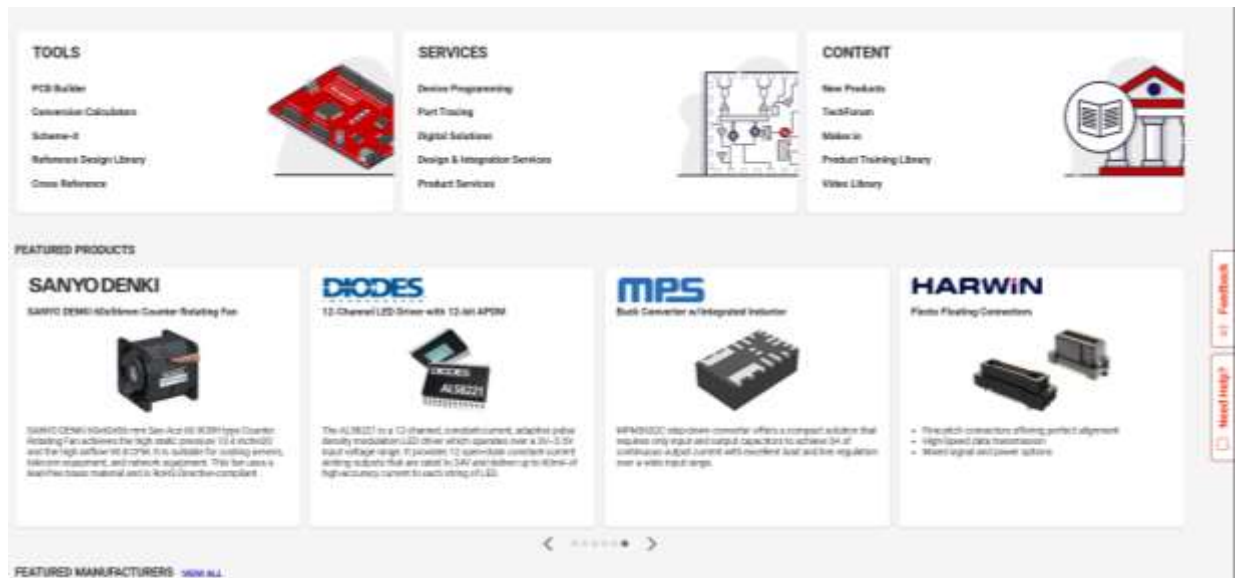


Рис. 1.15 – Головна сторінка сайту “DigiKey” (продовження)

Також, на відміну від деяких аналогічних ресурсів, окрім звичайного пошуку за ключовим словом чи номером, можна завантажити файл зі списком компонентів та провести пошук за допомогою цього файлу (рис.1.16). Авторизовані користувачі мають можливість зберегти список до улюблених та повернутися до нього за потреби.

File Upload Mapping

A preview of the upload is available below and columns have been mapped based on their contents. Please select the row where part info begins and verify that all columns are mapped appropriately.

ROW	COLUMN 1
☒ 1	Digi-Key Part Number
☐ 2	CS15186-12M

Continue

Рис. 1.16 – Пошук компонентів із завантаженого файлу на сайті “DigiKey”

При переході на сторінку категорії можна побачити детальний перелік фільтрів та варіантів сортування товарів (рис.1.17). Опису категорії, як на деяких подібних ресурсах, немає.

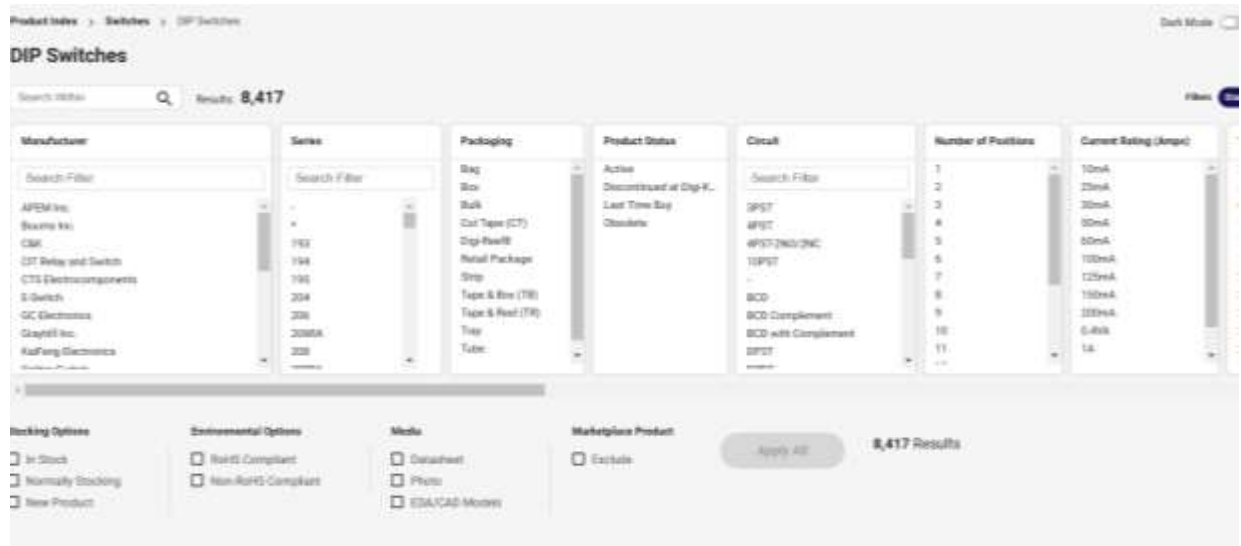


Рис. 1.17 – Сторінка категорії “Switches/DIP Switches” на сайті “DigiKey”

Нижче можна побачити перелік товарів, що належать до даної категорії та задовольняють обрані фільтри (рис. 1.18).

MFR Part #		Quantity Available		Price	Series	Package	Product Status	Circuit	Number of Positions	Current Rating (Amper)	Voltage Rating	Actuator Type	Actions
	 D501-254-L-010E SWITCH SLIDE DIP SPST 0.025A 24V <i>(Same Day (Formerly CJ) Delivery)</i>	13,732	In Stock	1 \$0.39000 Tube	D501-254	Tube 	Active	SPST	1	20mA	24VDC	Slide (Standard)	Raise
	 209-2M5 SWITCH SLIDE DIP SPST & 1A 30V <i>C75 Electronics Components</i>	41,868	In Stock	1 \$0.41000 Tube	209	Tube 	Active	SPST	2	100mA	20VDC	Slide (Standard)	Raise
	 210-2M5 SWITCH SLIDE DIP SPST & 1A 30V <i>C75 Electronics Components</i>	5,772	In Stock	1 \$0.40000 Tube	210	Tube 	Active	SPST	2	100mA	20VDC	Slide (Standard)	Raise
	 210-4M5 SWITCH SLIDE DIP SPST & 1A 30V <i>C75 Electronics Components</i>	4,215	In Stock	1 \$0.00000 Tube	210	Tube 	Active	SPST	4	100mA	20VDC	Slide (Standard)	Raise
	 219-2LPST SWITCH SLIDE DIP SPST & 1A 30V <i>C75 Electronics Components</i>	52,175	In Stock	1 \$0.50000 Tube	219	Tube 	Active	SPST	2	100mA	20VDC	Slide (Standard)	Raise
	 D654-254-L-010E-FAST	1,747	In Stock	1 \$0.57000 Tube	D654	Tube 	Active	SPST	0	20mA	24VDC	Slide	Raise

Рис. 1.18 – Перелік товарів окремої категорії та параметри сортування на сайті “DigiKey”

При переході на сторінку певного товару (рис. 1.19) можна побачити детальну інформацію про характеристики, ціни, наявність тощо, а також

перелік подібних товарів, що можуть зацікавити користувача.

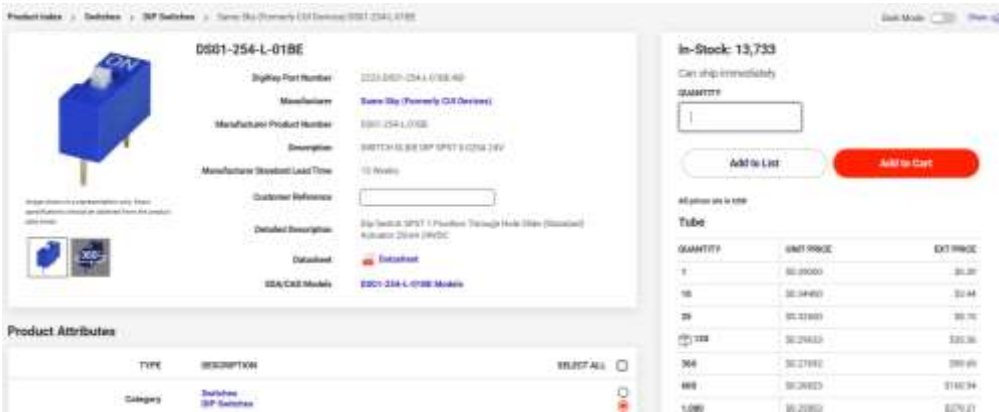


Рис. 1.19 – Сторінка окремого товару на сайті “DigiKey”

Для кількох обраних товарів можна створити порівняльну таблицю (рис.1.20). Таблиця враховує характеристики даних товарів, а для більшої зручності є опція «Приховати однакові атрибути».

2 Products		Hide Shared Attributes (13)
image		
Mfr Part #	209-2MS	DS01-254-L-01BE
Mfr	CTS Electrocomponents	Same Sky (Formerly CUI Devices)
Supplier	CTS Electrocomponents	Same Sky (Formerly CUI Devices)
DK Part #	CT2092MS-ND	2223-DS01-254-L-01BE-ND
Description	SWITCH SLIDE DIP SPST 0.1A 20V	SWITCH SLIDE DIP SPST 0.025A 24V
Price	\$0.41000	\$0.39000
Stock	41,868	13,733
Min Qty	1	1
Series	209	DS01-254
Package	Tube ⓘ	Tube ⓘ
Product Status	Active	Active

Рис. 1.20 – Порівняльна таблиця кількох компонентів на сайті “DigiKey”

Також сайт має типовий функціонал інтернет-магазину, такий як кошик користувача та система оплати товарів. Відповідно до наведеної вище інформації, можна виконати порівняння функціоналу сайтів “Otopart”, “Mouser Electronics” та “DigiKey” (таблиця 1.1).

Таблиця 1.1 — Порівняння перегляду даних в досліджених системах

Функціонал	Otopart	Mouser Electronics	Digikey
Головна сторінка	Пошук, перелік категорій, виробників, дистриб'юторів, популярні компоненти, новини зі сфери	Пошук, перелік категорій та виробників, перелік найновіших продуктів, обрані виробники, популярні категорії	Пошук, перелік категорій, виробників, обрані продукти, обрані виробники
Кількість продуктів у каталозі	Понад 72 млн компонентів від понад 400 дистриб'юторів та виробників	Понад 8 млн компонентів від дистриб'ютора Mouser Electronics	Понад 15 млн компонентів від понад 3000 постачальників
Меню категорій	Зручна, детальна ієрархія категорій та підкатегорій	Зручна, детальна ієрархія категорій та підкатегорій; наявна категорія «Нові продукти»	Зручна, детальна ієрархія категорій та підкатегорій; наявні категорії «Нові продукти» та «Без категорії»
Перегляд окремого продукту	Ціни та наявність в різних дистриб'юторів; технічні характеристики; подібні компоненти; детальний опис; зображення; інші можливі назви даного компонента	Ціна та наявність в дистриб'ютора “Mouser Electronics”; технічні характеристики; перелік схожих товарів	Ціна та наявність в дистриб'ютора “DigiKey”; технічні характеристики; перелік схожих товарів

У таблиці 1.2 описано порівняння функціоналу для аналізу даних, тобто функціоналу, який допомагає порівнювати та обирати конкретні продукти із загальної маси.

Таблиця 1.2 — Порівняння можливостей аналізу в досліджених системах

Функціонал	Octopart	Mouser Electronics	Digikey
Фільтрування продуктів	Різноманітні фільтри відповідно до типу продукту, а також фільтрування за дистриб'ютором	Різноманітні фільтри відповідно до типу продукту; фільтр «Новий продукт»	Різноманітні фільтри відповідно до типу продукту; фільтр «Новий продукт» та фільтри за доступністю зображення, схеми та CAD-моделі для електронних компонентів
Сортування продуктів	Сортування за різними параметрами відповідно до категорії	Сортування за різними параметрами відповідно до категорії	Сортування за різними параметрами відповідно до категорії
Порівняльна таблиця кількох продуктів	Можливість відсутня	Можна створити для будь-яких обраних компонентів	Можна створити для будь-яких обраних компонентів; можна приховати однакові атрибути
Пошук продуктів	За номером або за ключовим словом	За номером або за ключовим словом	За номером або за ключовим словом; пошук елементів зі списку з завантаженого текстового файлу

Отже, можна зробити висновок щодо функціоналу вебсайтів “Ostopart”, “Mouser Electronics”, “DigiKey”. Досліджені сайти мають багато спільного та

в основному мають одне і те ж призначення, однак мають деякі відмінності в будові головної сторінки, переліку категорій тощо.

Зокрема, найзручніший пошук доступний на сайті “DigiKey”, адже там додатково реалізована можливість пошуку списку елементів.

Перегляд продуктів окремої категорії найкраще реалізований на сайті “Mouser Electronics”, де реалізовано можливість завантажити таблицю з переліком, а також наявний опис категорії.

Найбільше фільтрів доступно на сайті “DigiKey”, де також наявні фільтри «Новий продукт», а також фільтри за наявністю медіа для компонентів (зокрема, зображень чи CAD-моделей).

Огляд окремого елемента є найбільш зручним на сайті “Octopart”. Зокрема, на цьому сайті можна переглянути можливі альтернативні назви того самого продукту. Тільки на сайтах “Mouser Electronics” та “DigiKey” є можливість створення порівняльної таблиці кількох елементів. У “DigiKey” також є можливість приховати однакові атрибути.

Головна відмінність полягає у тому, що “Octopart” не є продавцем електронних компонентів, а натомість дає можливість порівняти ціни та наявність в різних дистриб’юторів, що безумовно є зручним для користувача. Окрім того, така концепція дозволяє цьому сайту додавати до системи значно більшу кількість компонентів.

1.4. Постановка задачі

Інформаційна система для пошуку електронних компонентів повинна систематизувати електронні компоненти для забезпечення ефективного доступу до необхідних даних, спрощення процесу вибору та порівняння компонентів.

Для досягнення цієї мети система має підтримувати структуровану класифікацію компонентів, що враховує їхні технічні характеристики, категорії та виробників. Важливим аспектом є впровадження механізмів фільтрації та пошуку, які дозволяють користувачам швидко знаходити

						23

потрібні компоненти за заданими параметрами, такими як електричні характеристики, форма корпусу тощо. Система повинна обслуговувати два види користувачів: звичайних користувачів, які переглядають дані, та адміністраторів, які додатково можуть оновлювати дані в системі.

Для незареєстрованого або неавторизованого користувача необхідно реалізувати даний функціонал:

- 1) перегляд списку електронних компонентів з можливістю сортування, застосування фільтрів за різними параметрами;
- 2) перегляд списку категорій та виробників;
- 3) перегляд сторінки категорії із її коротким описом, переліком підкатегорій та переліком відповідних компонентів, які можна сортувати чи фільтрувати за бажанням користувача;
- 4) перегляд сторінки окремого компонента з його описом, характеристиками, фотографіями тощо;
- 5) пошук електронних компонентів за номером чи ключовими словами;
- 6) перегляд актуальних новин в сфері електроніки;
- 7) реєстрація в системі;
- 8) авторизація в системі.

Для зареєстрованого та авторизованого користувача необхідно реалізувати даний функціонал:

- 1) весь функціонал, доступний незареєстрованим та неавторизованим користувачам;
- 2) вихід з облікового запису;
- 3) додавання компонентів до списку улюблених;
- 4) відправлення скарги адміністраторам про некоректні дані в каталозі.

Для адміністратора необхідно реалізувати даний функціонал:

- 1) весь функціонал, доступний зареєстрованому та авторизованому користувачу;
- 2) додавання, редагування та видалення електронних компонентів або

						24

категорій із системи;

3) Перегляд отриманих скарг щодо некоректних даних в каталозі.

Нефункціональні вимоги до системи:

- 1) час завантаження сторінок не повинен перевищувати 2 секунд при середньостатистичному інтернет-з'єднанні;
- 2) захист від несанкціонованого доступу;
- 3) підтримка сучасних версій браузерів, таких як Google Chrome (версія 133+), Mozilla Firefox (версія 135+), Microsoft Edge (версія 132+), Safari (версія 18.3+).

Висновки до розділу 1

Перший розділ заклав теоретичну основу для розробки інформаційної системи, орієнтованої на пошук електронних компонентів, та обґрунтовує актуальність проєкту. Було здійснено огляд сучасного стану галузі електроніки та інформаційних технологій, визначено роль цифрових інструментів у прискоренні процесів розробки, конструювання та підбору компонентів.

Детально розглянуто основні принципи побудови інформаційних систем – зокрема, їх архітектуру, функціональні складові та життєвий цикл. Зроблено акцент на модульності, масштабованості, багаторівневому доступі до даних, що є важливими вимогами для систем, що працюють із великими обсягами технічної інформації та метаданих.

Важливою частиною стало дослідження існуючих рішень і аналогів. Було проаналізовано популярні платформи, такі як Ostopart, Digi-Key, FindChips, та визначено їхні переваги й обмеження. Це дало змогу виділити функціональні прогалини та сформувати унікальні вимоги до власної системи, які зумовили постановку задачі.

Було сформульовано мету створення системи – забезпечення швидкого, точного й зручного пошуку електронних компонентів з урахуванням технічних параметрів, виробника, наявності та ціни. Також були визначені

						25

ключові функціональні вимоги, що ляжуть в основу проєктування: підтримка фільтрації, інтеграція з зовнішніми базами, локальна індексація, адаптивний інтерфейс і підтримка аналітики.

інтегрувати машинне навчання [16].

PHP є однією з найдавніших і найпоширеніших мов для розробки динамічних вебсторінок. Вона добре інтегрується з HTML, і багато популярних систем управління контентом (наприклад, WordPress, Drupal, Joomla) написані саме на PHP. Ця мова відома своєю простотою у використанні для невеликих проєктів, але при цьому здатна обслуговувати великі системи за наявності правильної архітектури та використання сучасних фреймворків, таких як Laravel чи Symfony. PHP часто обирають для корпоративних сайтів, блогів, інтернет-магазинів і внутрішніх бізнес-додатків.

Java – це строго типізована об'єктно-орієнтована мова, яка традиційно використовується для розробки великих і надійних корпоративних кросплатформних вебдодатків. Вона має високу продуктивність і підтримує масштабування, що робить її популярною в банківському секторі, телекомунікаціях та великих комерційних системах. Завдяки платформі Spring Java дозволяє ефективно будувати backend логіку з чіткою структурою, безпечними транзакціями й розгалуженою системою контролю доступу.

C# є основною мовою для розробки вебдодатків у середовищі Microsoft. Використовується разом із фреймворком ASP.NET, який забезпечує високу продуктивність, гнучку архітектуру та глибоку інтеграцію з іншими продуктами Microsoft. C# часто обирають для корпоративних систем, внутрішніх порталів, бухгалтерських або управлінських рішень, особливо в організаціях, що вже використовують екосистему Microsoft (наприклад, SQL Server, Active Directory тощо).

Go – це мова, розроблена Google для створення швидких, надійних і паралельних систем. Вона особливо добре підходить для розробки серверних частин вебдодатків (backend), де важлива обробка великої кількості одночасних запитів. Go поєднує простий синтаксис із високою швидкістю виконання і має зручні інструменти для розгортання та підтримки сервісів.

Часто використовується у мікросервісній архітектурі, хмарних платформах і високонавантажених вебсервісах.

У випадку створення інформаційної системи для електронних компонентів мова Python є цілком обґрунтованим вибором. Вона поєднує простоту у використанні з широкими можливостями, що дозволяє зосередитися на логіці системи, а не на складнощах синтаксису чи низькорівневому програмуванні. Python має велику спільноту, активну екосистему бібліотек і готових рішень, які значно прискорюють розробку, особливо у сфері роботи з базами даних, API та обробки текстової інформації. Крім того, Python добре підходить для створення вебінтерфейсів і REST API, що є актуальним для систем, орієнтованих на перегляд, фільтрацію та пошук структурованих даних. Завдяки цьому мова Python дозволяє ефективно реалізувати функціональність, необхідну для інформаційного доступу до великої кількості компонентів, їх класифікації та зручного представлення користувачу.

Загалом Python є інтерпретованою, динамічно типізованою та об'єктно-орієнтованою мовою програмування. Його синтаксис максимально наближений до природної мови, що робить код читабельним та легким у підтримці. Python підтримує основні парадигми програмування – процедурну, об'єктно-орієнтовану та функціональну – завдяки чому залишається гнучким інструментом для вирішення різноманітних завдань. Однією з ключових особливостей є автоматичне управління пам'яттю та наявність вбудованих структур даних (словники, списки, множини, кортежі), що дозволяє швидко створювати складну логіку без зайвої технічної обробки низького рівня. Також Python підтримує структурований підхід до організації коду за допомогою модулів і пакетів, що забезпечує зручність повторного використання та розширення проєктів[15]. Усе це робить Python універсальним інструментом, придатним як для швидкого прототипування, так і для створення надійних промислових систем.

2.2. Вибір фреймворку

Фреймворк (від англ. framework – «каркас», «основа») у контексті програмування – це набір готових інструментів, бібліотек і правил, який допомагає розробникам швидше створювати програмне забезпечення, дотримуючись певної структури. Його основне призначення – спростити процес розробки, автоматизуючи типові задачі (наприклад, маршрутизацію запитів, роботу з базою даних чи формування HTML-відповідей) і забезпечити єдині підходи до організації коду. На відміну від звичайних бібліотек, які програміст викликає за потреби, фреймворк сам задає загальний хід виконання програми, дозволяючи розробникові вписувати власну логіку у визначені місця. Таким чином, фреймворк виступає як основа, на якій будується вся структура майбутнього застосунку.

Flask – це легкий та мінімалістичний вебфреймворк, який надає базові можливості для створення вебдодатків, залишаючи розробнику значну гнучкість у виборі архітектури, сторонніх бібліотек і способів реалізації функціоналу. Завдяки цьому Flask часто використовують у невеликих проєктах, стартапах, експериментальних розробках або для створення REST API. До переваг фреймворку належать простота освоєння, хороша документація та велика спільнота. Водночас його головним недоліком є те, що при зростанні складності проєкту розробникові доводиться самостійно вирішувати багато архітектурних питань, що може призвести до неузгодженості та складнощів з підтримкою[10].

FastAPI – сучасний асинхронний вебфреймворк, орієнтований на створення високопродуктивних REST API з автоматичною генерацією документації. Його ключові переваги – підтримка типізації, висока швидкість роботи завдяки асинхронності та зручна інтеграція зі стандартом OpenAPI. FastAPI ідеально підходить для мікросервісів, інтеграції з машинним навчанням або систем, де критична продуктивність. Основний недолік – відносна новизна, а отже, менша кількість прикладів і гірша сумісність з деякими старішими бібліотеками порівняно з класичними фреймворками [9].

						30

Django – це потужний і повнофункціональний фреймворк, який реалізує принцип "батареї в комплекті" (batteries included), тобто постачається з великою кількістю вбудованих інструментів для типових завдань: аутентифікації, роботи з базою даних, адміністрування, форм тощо. Він ідеально підходить для інформаційних систем, корпоративних вебдодатків, новинних порталів, маркетплейсів та інших проєктів, які потребують чіткої структури та швидкого запуску. Серед його переваг – зрозуміла архітектура (MTV, або Model-Template-View, тобто Модель-Шаблон-Представлення), висока якість документації, активна спільнота та зручна система розширень [6]. Недоліком може бути певна "жорсткість" – Django задає багато стандартів і підходів, що обмежує гнучкість на складних або нестандартних проєктах. Тим не менш, саме ця структурованість часто вважається перевагою в довгостроковій перспективі, особливо у командній розробці.

У контексті розробки інформаційної системи для електронних компонентів Django виявився найбільш доцільним вибором завдяки своїй структурованості, наявності готових інструментів і зручній роботі з реляційними базами даних. Проєкт передбачає зберігання великої кількості даних із чіткою структурою – категорії, параметри, виробники, а також підтримку функцій пошуку, фільтрації та представлення інформації в зручному форматі. Django надає всі необхідні засоби для реалізації таких вимог "із коробки": ORM (Object Relation Mapping, або об'єктно-реляційне відображення) для взаємодії з базою даних, зручну систему шаблонів, автоматично згенеровану адміністративну панель, а також добре продуману архітектуру, яка спрощує подальшу підтримку і масштабування проєкту. Завдяки цьому розробка системи проходить швидше, а результат – більш стабільний і готовий до розширення у майбутньому.

Однією з найпотужніших складових Django є його система ORM (Object-Relational Mapping) – механізм, що дозволяє працювати з базою даних через об'єкти Python, без потреби писати SQL-запити вручну. Усе, що

потрібно – визначити моделі (класи), які відповідають таблицям у базі, а фреймворк сам створить необхідні таблиці, зв'язки між ними та забезпечить засоби для запитів, фільтрації, оновлення та видалення даних. Це значно підвищує читабельність і безпечність коду, оскільки дозволяє уникнути ручної обробки SQL-запитів і зменшує ризик помилок, наприклад, ін'єкцій.

Основу структури додатка в Django складає архітектурний підхід Model-Template-View (MTV), який є варіацією класичної моделі MVC. У цьому підході Model відповідає за логіку збереження даних, Template – за відображення HTML-інтерфейсу, а View – за логіку обробки запитів і підготовку відповідей. Такий поділ дає змогу чітко відокремити відповідальність між різними частинами системи, полегшує масштабування проєкту та сприяє кращій організації коду.

Ще однією важливою перевагою Django є вбудована адміністративна панель, яка генерується автоматично на основі визначених моделей. Вона надає зручний графічний інтерфейс для перегляду, редагування й управління даними без написання додаткового коду. Адмін-панель легко налаштовується, підтримує пошук, фільтрацію, сортування, а також керування правами доступу користувачів. Завдяки цьому навіть складні системи з великою кількістю сутностей можуть обслуговуватись адміністраторами без потреби залучення розробника.

Окрім основних компонентів, Django містить безліч вбудованих рішень для типових завдань. Зокрема, фреймворк включає систему обробки форм, засоби валідації даних, механізми керування сесіями, інструменти для безпечної аутентифікації та авторизації користувачів, систему перекладів, кешування, захист від CSRF-атак тощо. Така комплектація робить Django особливо привабливим для проєктів, де важлива безпека, стабільність і швидкість розробки.

2.3. Вибір СУБД

База даних – це організоване сховище інформації, яке дозволяє зберігати, шукати, оновлювати та керувати даними ефективно та надійно. Вона є центральним елементом багатьох сучасних програмних систем, незалежно від їх масштабу – від невеликих вебдодатків до складних корпоративних платформ. У системах, де обробляється велика кількість структурованих або частково структурованих даних, база даних забезпечує цілісність, збереження та доступність інформації. Саме вона дозволяє розмежовувати логіку обробки даних від їх фізичного зберігання, що робить систему гнучкішою, зручнішою для підтримки й масштабування. У проєктах, пов'язаних з обліком, каталогізацією або фільтрацією об'єктів, таких як інформаційні системи, роль бази даних є ключовою – без неї неможливо ефективно реалізувати навіть базові функції.

Система управління базами даних (СУБД) – це програмне забезпечення, яке забезпечує створення, підтримку та використання бази даних. Вона виконує роль посередника між користувачем (або застосунком) і самою базою даних, надаючи інтерфейси для зберігання, модифікації, пошуку й видалення даних. СУБД автоматизує багато важливих процесів – від контролю цілісності даних і керування транзакціями до забезпечення багатокористувацького доступу й захисту інформації. Завдяки СУБД розробники можуть зосередитись на бізнес-логіці застосунку, не переймаючись деталями зберігання або оптимізації доступу до даних на фізичному рівні. Існує багато різних СУБД, кожна з яких має свої особливості, підходи до організації даних і сфери застосування.

Системи управління базами даних (СУБД) поділяються на два основні класи – реляційні та нереляційні, залежно від способу організації, зберігання й обробки інформації. Вибір між цими двома підходами має суттєвий вплив на архітектуру програмного забезпечення, його продуктивність, можливості масштабування та підтримки. Кожен з варіантів має свої переваги і недоліки, тому під час проєктування будь-якої інформаційної системи важливо

						33

враховувати, який тип бази найкраще відповідає вимогам проєкту.

Реляційні бази даних використовують табличну модель для представлення інформації, де дані організовані у вигляді таблиць, що мають фіксовану структуру – визначені поля (стовпці) з типами даних. Ключовою особливістю цього підходу є підтримка чітко описаних зв'язків між таблицями за допомогою зовнішніх ключів. Реляційні СУБД забезпечують узгодженість, цілісність та нормалізацію даних, дозволяючи уникати дублювання та логічних суперечностей. Завдяки підтримці транзакцій, системи цього типу гарантують надійність при одночасному доступі кількох користувачів. Мова SQL (Structured Query Language) використовується для створення, зміни, вибірки та аналізу даних, що дає широкі аналітичні можливості.

До переваг реляційного підходу також належить чітка схема даних, що дає змогу контролювати структуру та очікувану якість записів. Проте така строгість може стати недоліком у випадках, коли структура даних швидко змінюється або наперед важко передбачити всі типи зв'язків і атрибутів. Крім того, масштабування реляційних СУБД у горизонтальному напрямі (на кілька серверів) є складним завданням, що обмежує їх застосування в деяких високонавантажених проєктах. Втім, для систем із чітко визначеною структурою – таких як фінансові сервіси, управління запасами або інформаційні каталоги – реляційні бази залишаються стандартом де-факто [21].

Нереляційні бази даних (NoSQL) пропонують альтернативу табличній моделі й орієнтовані на зберігання неструктурованих або напівструктурованих даних. Вони включають кілька підтипів: документні бази (наприклад, MongoDB), бази типу ключ-значення (Redis), графові бази (Neo4j) та сховища з широким стовпчиком (Cassandra). Такі СУБД зазвичай не потребують попереднього визначення схеми, що дозволяє більш гнучко змінювати структуру даних. Це особливо корисно у стартап-проектах або в системах, що швидко еволюціонують.

Однією з ключових переваг NoSQL є простота масштабування в горизонтальному напрямі, що робить ці рішення популярними для великих вебсервісів і застосунків із мільйонами користувачів, як-от соціальні мережі, стрімінгові платформи чи системи аналітики в реальному часі. З іншого боку, NoSQL-бази не завжди забезпечують такий самий рівень надійності транзакцій і гарантій цілісності даних, як реляційні системи, що робить їх менш придатними для застосунків, де помилки в даних є критичними[20].

Rank			DBMS	Database Model	Score		
May 2025	Apr 2025	May 2024			May 2025	Apr 2025	May 2024
1.	1.	1.	Oracle	Relational, Multi-model 	1226.57	-4.49	-9.72
2.	2.	2.	MySQL	Relational, Multi-model 	964.98	-22.12	-118.76
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model 	774.89	-10.12	-49.41
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	674.32	+7.07	+28.77
5.	5.	5.	MongoDB 	Document, Multi-model 	402.51	+2.46	-19.14
6.	6.	 9.	Snowflake	Relational	172.01	+3.92	+50.67
7.	7.	 6.	Redis	Key-value, Multi-model 	152.19	-1.92	-5.61
8.	 9.	8.	IBM Db2	Relational, Multi-model 	126.40	+0.36	-2.06
9.	 8.	 7.	Elasticsearch	Multi-model 	123.81	-4.27	-11.54
10.	10.	10.	SQLite	Relational	117.77	+3.68	+3.45

Рис. 2.1 — Найбільш поширені СУБД станом на травень 2025 року

На рисунку 2.1 показано рейтинг найбільш поширених СУБД станом на травень 2025 року [3]. У цьому рейтингу можна виділити як реляційні, так і нереляційні рішення. Серед реляційних лідерами залишаються MySQL, PostgreSQL та SQLite. MySQL відомий своєю простотою, швидкістю і широкою підтримкою хостингів, тому часто використовується у веброзробці. PostgreSQL, у свою чергу, виділяється розширеними можливостями, підтримкою складних запитів, транзакцій і розширень, що робить його популярним вибором для складних систем. SQLite – це вбудована СУБД, яка не потребує окремого серверного оточення й ідеально підходить для легких або автономних застосунків. Серед нереляційних СУБД найбільш відомими є MongoDB, яка використовує документну модель зберігання, та Redis, що працює як швидке сховище типу ключ-значення і часто використовується для

кешування або обробки тимчасових даних. Вибір конкретної СУБД залежить від типу даних, функціональних вимог системи та потреб у масштабуванні.

У випадку розробки інформаційної системи для електронних компонентів використання реляційної бази даних є найбільш доцільним через потребу в чітко структурованих даних та складних взаємозв'язках між різними об'єктами. Система зберігає багато типів інформації, таких як компоненти, категорії, технічні характеристики, виробники та користувачі, які мають взаємозалежності. Наприклад, кожен компонент належить до певної категорії, має множину характеристик, пов'язаний з виробником і може бути доданий до списку улюблених користувачем. Це вимагає використання реляційної моделі для зберігання даних у вигляді таблиць, де чітко визначені зв'язки між сутностями. Така організація дозволяє легко підтримувати цілісність даних, забезпечуючи їх точність і відповідність бізнес-логіці системи.

SQLite – легка реляційна СУБД, яка працює як вбудоване рішення без необхідності окремого сервера. Вона часто використовується в мобільних додатках, а також у десктопних програмах, де немає потреби в великому обсязі даних або високих вимогах до масштабування. Її основною перевагою є простота налаштування та використання, а також малий розмір і низькі вимоги до ресурсів. Проте SQLite не підходить для масштабованих вебдодатків з великою кількістю одночасних користувачів, оскільки не підтримує розподілені системи і має обмеження на одночасний доступ до бази даних. Ця СУБД найбільш ефективно використовується для локальних застосунків, які не потребують складної серверної архітектури[18].

MySQL – одна з найпоширеніших реляційних СУБД, яка здобула популярність завдяки своїй простоті, швидкості та широкій підтримці на різних хостинг-платформах. Вона має відкритий код і добре підходить для середніх і малих вебдодатків, де не потрібно обробляти дуже складні запити або великий обсяг транзакцій. До переваг можна віднести простоту налаштування, активну спільноту та зручну документацію, що забезпечує

швидке вирішення проблем. Однак у MySQL можуть виникати обмеження на складність запитів, а також відсутність деяких функцій, які доступні в інших СУБД, наприклад, у PostgreSQL. Тому вона найкраще підходить для застосунків із помірною складністю запитів та середнім навантаженням[12].

PostgreSQL – потужна реляційна СУБД, відома своєю стабільністю, масштабованістю та підтримкою розширених можливостей, таких як складні запити, збережені процедури та транзакції. Вона також підтримує індексацію за допомогою різноманітних методів, що дозволяє ефективно працювати з великими обсягами даних. PostgreSQL має безліч вбудованих функцій для роботи з географічними даними (PostGIS) та підтримує складніші типи даних, ніж MySQL. Однак її налаштування може бути складнішим, а продуктивність може бути нижчою для простих запитів порівняно з MySQL. PostgreSQL ідеально підходить для великих проєктів, де необхідна висока надійність, безпека і можливість розширення, таких як фінансові системи або аналітичні платформи[14].

Для розробки інформаційної системи електронних компонентів підійде СУБД PostgreSQL, оскільки вона найкраще відповідає вимогам інформаційної системи для електронних компонентів. Завдяки підтримці складних зв'язків між таблицями, надійній роботі з транзакціями, розширеним можливостям SQL і високій стабільності PostgreSQL забезпечує ефективне зберігання й обробку структурованих даних. Вона також дозволяє зручно реалізовувати фільтрацію, сортування, агрегацію та інші запити, необхідні для побудови каталогу компонентів і роботи з обліковими записами користувачів. Крім того, PostgreSQL має активну спільноту та хорошу інтеграцію з фреймворком Django, що додатково спрощує розробку й підтримку системи.

Висновки до розділу 2

У другому розділі проведено обґрунтований вибір технологій, інструментів і платформ, які використовуватимуться для реалізації

						37

інформаційної системи пошуку електронних компонентів. Підібрані засоби відповідають вимогам до продуктивності, масштабованості, зручності підтримки та інтеграції з зовнішніми джерелами даних.

Розглянуто доступні мови програмування з урахуванням їхньої екосистеми, підтримки веброзробки, швидкодії та доступності бібліотек. Було обрано мову Python як базову для серверної частини проєкту завдяки її читабельності, широкому застосуванню в розробці веб-API та великій кількості бібліотек для роботи з даними.

Аргументовано вибір FastAPI як основного фреймворку для створення серверної логіки. Цей фреймворк забезпечує високу швидкість обробки запитів, має вбудовану підтримку OpenAPI документації, легко масштабується та добре підходить для реалізації RESTful-сервісів, які є основою для роботи з даними електронних компонентів.

Щодо зберігання даних, розглянуто кілька популярних систем управління базами даних. У результаті було обрано PostgreSQL як основну СУБД завдяки її надійності, підтримці складних запитів, розширень (наприклад, для роботи з JSON), масштабованості та активній спільноті. Альтернативно, для локального тестування або мобільного прототипування також може використовуватись SQLite.

Таким чином, обрані інструменти дозволяють створити гнучку, продуктивну та масштабовану інформаційну систему, яка буде зручною як для користувача, так і для подальшого розвитку чи інтеграції з іншими сервісами.

3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Створення моделі «сутність-зв'язок»

База даних є основою інформаційної системи для пошуку електронних компонентів, оскільки саме вона забезпечує зберігання, впорядкування та доступ до усіх ключових даних про компоненти, їх характеристики, категорії, виробників і користувачів системи. Її структура повинна бути ретельно спроектована, щоб забезпечити зручність пошуку, фільтрації та масштабування в майбутньому. Важливо досягти високого ступеня нормалізації, що дозволить уникнути надлишковості, забезпечити цілісність даних і полегшити супровід системи. Крім того, база даних повинна точно відображати логічні зв'язки між об'єктами предметної області, бути узгодженою, гнучкою до змін і придатною для розширення функціональності. Метою проєктування є створення чіткої, логічної та масштабованої моделі, яка відповідатиме як поточним вимогам застосування, так і майбутнім сценаріям розвитку.

Першим етапом у проєктуванні бази даних є виділення сутностей. Сутність – це будь-який об'єкт або концепція, яка має значення в контексті бази даних і яку необхідно зберігати. Це можуть бути реальні об'єкти (наприклад, електронні компоненти, користувачі) або абстракції (наприклад, категорії). Сутність зазвичай відображається як таблиця в реляційній базі даних. Кожна сутність повинна бути описана за допомогою атрибутів, які відображають її характеристики, а також повинна бути визначена взаємодія з іншими сутностями через зв'язки, що забезпечить логічну організацію даних[8].

Для реалізації основного функціоналу каталогу необхідно використати такі сутності, як *Електронний Компонент*, *Категорія*, а також *Користувач* та *Скарга*. Для реалізації додаткового функціоналу, а саме перегляду останніх новин зі сфери електроніки, потрібна сутність *Новина*.

Другим етапом є визначення основних атрибутів для кожної сутності.

Наприклад, сутність *Електронний Компонент* має такі атрибути як назва, виробник, опис, зображення, а також атрибути для технічних характеристик, такі як тип корпусу, робоча напруга, робочий струм, потужність.

Сутність *Категорія* має такі основні атрибути як назва та опис. Ця сутність необхідна для систематизації компонентів у системі.

Сутність *Користувач* має такі атрибути як логін та пароль, а також роль (наприклад, простий користувач чи адміністратор).

Сутність *Скарга* допомагає користувачам повідомляти адміністраторів про неповні або некоректні дані в системі. Отже, ця сутність повинна мати такі основні атрибути, як опис, тобто опис проблеми, дата додавання, щоб адміністраторам було зручніше переглядати скарги, а також статус (наприклад, «оброблено» чи «в обробці», щоб адміністратори могли бачити, які скарги ще потребують розгляду).

Сутність *Новина* відображає конкретну статтю із зовнішнього ресурсу. В даній системі ця сутність повинна мати такі атрибути, як назва новини, посилання на оригінал статті, скорочена версія (частина оригінального тексту, що буде відображатися користувачам вебдодатку до переходу на джерело), обкладинка статті та дата додавання новини.

Третім етапом у проектуванні бази даних є визначення зв'язків між сутностями. Зв'язки дозволяють визначати, як сутності взаємодіють між собою.

Існує три основних видів зв'язків між сутностями. Зв'язок «один до одного» виникає, якщо кожен запис із таблиці А пов'язаний із одним і тільки одним об'єктом із таблиці Б. Зв'язок «один до багатьох» означає, що один запис в таблиці А може бути пов'язаний із кількома записами в таблиці Б. Зв'язок «багато до багатьох» існує, якщо один запис із таблиці А може бути пов'язаний із кількома записами із таблиці Б, і навпаки, один запис із таблиці Б може бути пов'язаний із кількома записами із таблиці А.

Для відображення сутностей та зв'язків між ними можна побудувати модель «сутність-зв'язок», або Entity-relationship model. Одним із способів

зображення такої моделі є нотація «вороняча лапка», або “crow’s foot”. Тоді сутності відображаються прямокутниками, а зв’язки – лініями. Тип зв’язку також відображається графічно.

Зв’язки між сутностями у даній інформаційній системі зображено на рисунку 3.1.

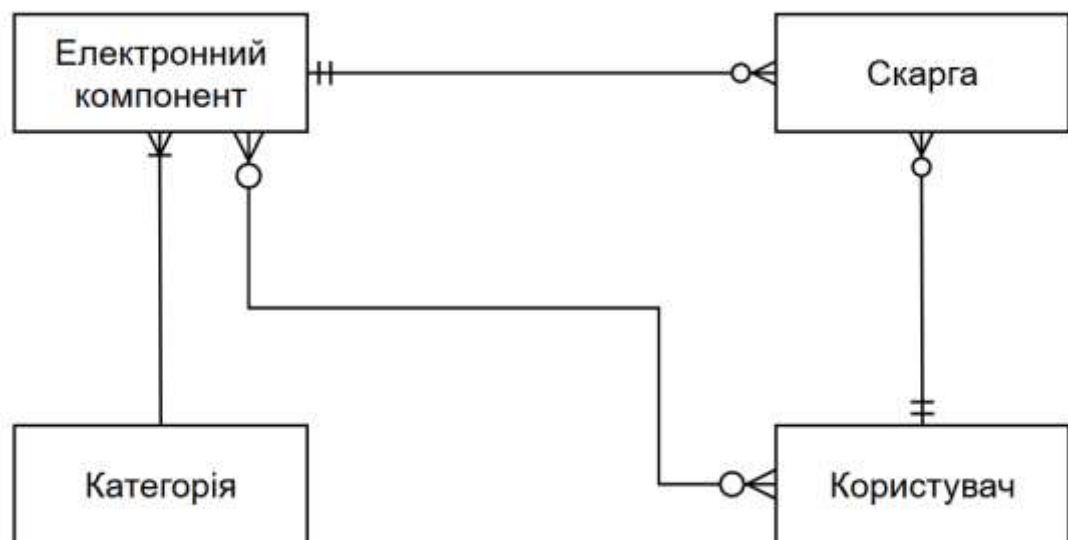


Рис. 3.1 — Модель «сутність-зв’язок»

У моделі представлено такі зв’язки як Компонент-Категорія, Скарга-Користувач, Скарга-Компонент та Компонент-Користувач.

Зв’язок між сутностями *Категорія* та *Електронний Компонент* – «один до багатьох». Це означає, що один компонент належить до однієї категорії, а одна категорія може містити багато компонентів.

Зв’язок між сутностями *Скарга* та *Користувач* – «один до багатьох». Кожна скарга обов’язково має одного автора (користувача), а кожен користувач може подати багато скарг, або не подати жодної.

Зв’язок між сутностями *Електронний Компонент* та *Скарга* – «один до багатьох». Тобто, кожна скарга стосується одного і тільки одного компонента, а на один компонент можна подати безліч скарг, або не подавати жодної.

Зв’язок між сутностями *Електронний Компонент* та *Користувач*

необхідний, щоб користувачі мали можливість додавати компоненти до списку улюблених. Отже, потрібен зв'язок «багато-до-багатьох». Тоді один користувач може мати багато улюблених компонентів (або не мати жодного), а один компонент може належати до улюблених для багатьох користувачів (або не належати до списку улюблених жодного користувача).

Сутність *Новина* не відображена на схемі, оскільки у даній системі немає необхідності у зв'язках цієї сутності з іншими.

3.2. Проектування бази даних

Після створення моделі «сутність-зв'язок», можна переходити до проектування бази даних у вигляді таблиць. Необхідно перетворити сутності у таблиці, атрибути – у назви полів, визначити типи та розмір полів, а також визначити ключі у таблицях. Важливо також визначити обов'язковість заповнення полів, унікальні обмеження, значення за замовчуванням та необхідні індекси [19]. Такий етап дозволяє створити повноцінну фізичну модель даних, яку надалі можна реалізувати за допомогою ORM у вибраному фреймворку.

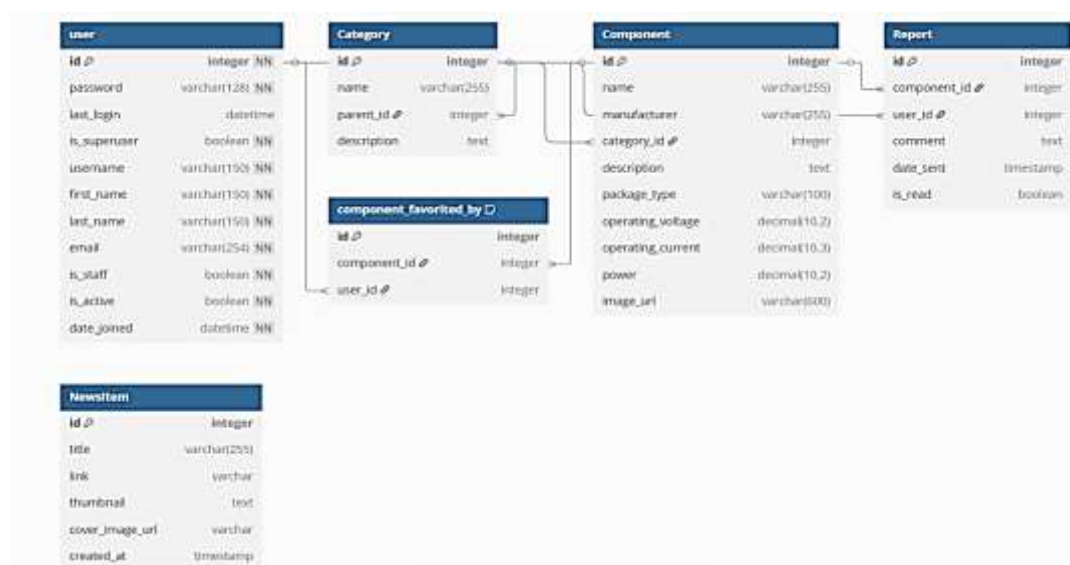


Рис. 3.2 — Загальна схема бази даних

На рисунку 3.2 зображена загальна схема бази даних, що складатиметься із

шести таблиць. Таблиці user, Category, Component, Report, NewsItem відображають сутності *Користувач*, *Категорія*, *Компонент*, *Скарга* та *Новина* відповідно, а component_favorited_by – додаткова таблиця для реалізації зв'язку «багато-до-багатьох».

У кожній із таблиць є унікальне поле id, яке є первинним ключем і дозволяє відрізнити окремі записи. Це поле завжди має цілочисельне значення, яке збільшується для кожного наступного запису.

Для реалізації зв'язків між таблицями існують зовнішні ключі. Зовнішній ключ у таблиці вказує на первинний ключ іншої таблиці та гарантує, що значення цього поля або дорівнює наявному значенню в пов'язаній таблиці, або є порожнім (у випадку необов'язкового зв'язку). Зовнішні ключі дозволяють моделювати зв'язки «один-до-багатьох», «багато-до-одного» або «багато-до-багатьох» та забезпечують збереження цілісності при оновленні або видаленні пов'язаних даних.

Таблиця Component відповідає сутності *Електронний Компонент* і містить такі колонки:

- id (ідентифікатор) – первинний ключ, ціле число;
- name (назва) – рядок змінної довжини, максимальний розмір 255 символів;
- manufacturer (виробник) – рядок змінної довжини, максимальний розмір 255 символів;
- category_id (ідентифікатор категорії) – зовнішній ключ до таблиці Category; містить первинний ключ відповідного запису-категорії;
- description (опис) – текстове поле для опису компонента;
- package_type (тип корпусу) – рядок змінної довжини, максимальна довжина – 100 символів;
- operating_voltage (робоча напруга, V) - десяткове число з точністю до 2 знаків після коми, максимальна довжина – 10 цифр;
- operating_current (робочий струм, A) – десяткове число з точністю до 3 знаків після коми, максимальна довжина – 10 цифр;

- power (потужність, W) – десяткове число з точністю до 2 знаків після коми, максимальна довжина – 10 цифр;
- image_url (посилання на зображення) – рядок до 600 символів.

Отже, у таблиці більшість колонок збігаються з визначеними раніше атрибутами сутності, а також додаються зовнішній ключ до *Категорії* та первинний ключ. Також для кожної колонки є визначений тип та максимальна довжина, що дозволяє оптимізувати зберігання даних.

Таблиця Category відповідає сутності *Категорія* і містить такі колонки:

- id (ідентифікатор) - первинний ключ, ціле число;
- name (назва) – рядок змінної довжини, максимальний розмір 255 символів;
- parent_id (батьківський ідентифікатор) – зовнішній ключ на Category.id, дозволяє створювати підкатегорії. Може бути порожнім, якщо батьківської категорії немає;
- description (опис) – текстове поле для опису категорії.

Аналогічно до попередньої, ця таблиця містить колонки, що збігаються з атрибутами відповідної сутності, а також первинний ключ та зовнішній ключ для побудови ієрархії категорій.

Таблиця Report відповідає сутності *Скарга* та містить такі колонки:

- id (ідентифікатор) - первинний ключ, ціле число;
- component_id (ідентифікатор компонента) - зовнішній ключ на Component.id, представляє компонент, на який подано скаргу;
- user_id (ідентифікатор користувача) – зовнішній ключ на User.id, представляє користувача, який подав скаргу;
- comment (коментар) – текстове поле, коментар від користувача із поясненням суті скарги;
- date_sent (дата відправлення) – дата та час створення звіту, встановлюється автоматично при додаванні скарги;
- is_read (чи прочитано) – статус скарги (прочитана адміністратором чи ні), булеве поле.

						44

Отже, таблиця Report має колонки, що відповідають атрибутам сутності *Скарга*, а також первинний ключ та зовнішні ключі для зв'язку з *Користувачем* та *Компонентом*.

Для представлення сутності Користувач у даному проєкті достатньо використати типову модель користувача Django (`django.contrib.auth.models.User`), оскільки вона забезпечує повний набір базової функціональності для автентифікації та керування користувачами: зберігання імені користувача, електронної пошти, пароля, статусу активності, прав доступу тощо. Ця модель добре інтегрується з адмін-панеллю, системою авторизації та дозволів, а також підтримується багатьма сторонніми бібліотеками.

Враховуючи, що в межах проєкту не було потреби в розширених полях профілю користувача (наприклад, аватар чи біографія), використання стандартної моделі є найпростішим і надійним рішенням, яке дозволяє уникнути надлишкової складності та зосередитися на основній бізнес-логіці застосунку.

Таблиця, що відповідає стандартній моделі користувача Django, містить такі колонки:

- `id` (ідентифікатор) - первинний ключ, ціле число;
- `password` (пароль) – рядок змінної довжини, максимальна довжина – 128 символів;
- `last_login` (останній вхід) – дата та час останнього входу в обліковий запис;
- `is_superuser` (чи є суперкористувачем) – булеве поле, використовується для визначення додаткових прав користувача;
- `username` (ім'я користувача) – рядок змінної довжини, максимальна довжина – 150 символів;
- `first_name` (ім'я) – рядок змінної довжини, максимальна довжина – 150 символів;
- `last_name` (прізвище) – рядок змінної довжини, максимальна

						45

довжина – 150 символів;

- email (електронна пошта) – рядок змінної довжини, максимальна довжина – 254 символи;
- is_staff (чи є працівником) – булеве значення, використовується для визначення додаткових прав користувача;
- is_active (чи є активним) – булеве значення, використовується для визначення прав користувача;
- date_joined (дата приєднання) – дата та час реєстрації користувача.

Отже, стандартна модель користувача Django повністю покриває вимоги даного проєкту.

Таблиця component_favorited_by – зв’язувальна таблиця для реалізації зв’язку «багато-до-багатьох», що виникає автоматично при створенні такого зв’язку через Django. Дана таблиця використовується для додавання електронних компонентів до списку улюблених. Необхідні поля – лише власний первинний ключ та первинні ключі на Component.id та User.id.

Таблиця NewsItem представляє сутність *Новина* та містить такі колонки:

- id (ідентифікатор) - первинний ключ, ціле число;
- title (заголовок) – рядок змінної довжини, максимальна довжина 255;
- link (посилання) – посилання на оригінал статті, рядок змінної довжини;
- thumbnail (початок) – текстове поле, перший абзац оригінальної статті, що буде відображатися у додатку;
- cover_image_url (посилання на зображення обкладинки) – рядок змінної довжини, посилання на обкладинку оригінальної статті;
- created_at (час додавання) – дата та час додавання новини.

Отже, дана таблиця містить колонки, що відповідають атрибутам сутності *Новина*, а також первинний ключ. Зовнішніх ключів немає, оскільки немає потреби в створенні зв’язків з іншими таблицями.

						46

3.3. Визначення структури проєкту Django

У Django проєкт – це загальна структура вебзастосунку, що об'єднує налаштування, конфігурацію і набір функціональних компонентів. Один Django-проєкт містить головний конфігураційний файл `'settings.py'`, де визначаються всі ключові параметри, включаючи встановлені додатки (apps), налаштування бази даних, маршрутизацію, шаблони тощо.

Django-проєкт не є самодостатнім функціональним модулем, а скоріше контейнером для окремих додатків, кожен з яких відповідає за певну частину логіки застосунку.

Django-додаток (app) – це незалежний модуль всередині проєкту, який виконує певну функцію. Кожен додаток має власну структуру: файли моделей, представлень (views), шаблонів, форм, URL-маршрутів тощо. Django дозволяє легко повторно використовувати додатки в різних проєктах, якщо вони достатньо універсальні.

Проєкти зазвичай розділяють на додатки відповідно до логічних функціональних блоків системи. Такий поділ спрощує підтримку, масштабування та тестування коду. Важливо дотримуватись принципу чіткого розмежування відповідальностей: кожен додаток має мати чітко визначену функцію і не дублювати логіку інших[4].

У даному проєкті варто виділити такі три додатки як catalog (каталог), users (користувачі) та news (новини).

У такому разі додаток catalog відповідає за основний функціонал проєкту, що пов'язаний із електронними компонентами. Зокрема, сюди входить обробка компонентів, категорій та скарг. Компоненти, категорії та скарги зберігаються в базі даних. Користувачам доступні можливості для перегляду, пошуку, фільтрування, сортування компонентів, створення порівняльних таблиць та списків із компонентами, а також подання скарг. Адміністраторам доступні можливості зміни, додавання чи видалення компонентів або категорій, а також перегляд та обробка скарг. Відповідно, в додатку catalog необхідно описати моделі *Компонент*, *Скарга* та *Категорія*, а

						47

також реалізувати представлення та шаблони для описаного вище функціоналу.

Додаток `users` відповідає за користувачів, авторизацію та автентифікацію.

Такі дії, як вхід чи вихід з облікового запису, реєстрація, перевірка статусу користувача (наприклад, адміністратор чи простий користувач) – виконуються саме через цей додаток. Django має багато зручних вбудованих механізмів для авторизації та автентифікації, таких як стандартна модель користувача та інші. Отже, у цьому додатку потрібно описати всю логіку, що пов'язана із користувачами.

Додаток `news` у проєкті відповідає за збирання, зберігання та відображення користувачам новин, пов'язаних з інженерією, електронікою та електронними компонентами.

Його головне призначення – автоматично отримувати актуальні публікації з зовнішніх джерел (наприклад, новинних сайтів або блогів виробників) за допомогою механізму вебскрейпінгу. Отримані новини зберігаються в базі даних, після чого можуть відображатися у зручному форматі для користувачів системи. Така функціональність допомагає тримати користувачів в курсі останніх подій у сфері електроніки, доповнюючи основну інформаційну систему динамічним контентом. Отже, у цьому додатку потрібно описати модель *Новина*, а також реалізувати збирання, зберігання та відображення новин для користувачів.

3.4. Розробка інформаційної системи

На рисунку 3.3 представлено реалізацію сутності *Електронний Компонент*:

```

class Component(models.Model):
    name = models.CharField(max_length=255)
    manufacturer = models.CharField(max_length=255)
    category = models.ForeignKey('Category', on_delete=models.CASCADE)
    description = models.TextField(null=True, blank=True)
    favorited_by = models.ManyToManyField(User, related_name='favorite_components', blank=True)

    package_type = models.CharField("Тип корпусу", max_length=100, null=True, blank=True)
    operating_voltage = models.DecimalField("Робоча напруга (V)", max_digits=10, decimal_places=2, null=True, blank=True)
    operating_current = models.DecimalField("Робочий струм (A)", max_digits=10, decimal_places=3, null=True, blank=True)
    power = models.DecimalField("Потужність (W)", max_digits=10, decimal_places=2, null=True, blank=True)
    image_url = models.URLField("Зображення (URL)", max_length=600, null=True, blank=True)

```

Рис. 3.3 — Реалізація сутності *Електронний Компонент*

На рисунку 3.4 представлено реалізацію сутності *Категорія*:

```

class Category(models.Model):
    name = models.CharField(max_length=255)
    parent = models.ForeignKey('self', null=True, blank=True, on_delete=models.CASCADE)
    description = models.TextField(null=True, blank=True)

```

Рис. 3.4 — Реалізація сутності *Категорія*

На рисунку 3.5 представлено реалізацію сутності *Скарга*:

```

class Report(models.Model):
    component = models.ForeignKey("Component", on_delete=models.CASCADE, related_name="reports")
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    comment = models.TextField()
    date_sent = models.DateTimeField(auto_now_add=True)
    is_read = models.BooleanField(default=False)

```

Рис. 3.5 — Реалізація сутності *Скарга*

```

class NewsItem(models.Model):
    title = models.CharField(max_length=255)
    link = models.URLField()
    thumbnail = models.TextField()
    cover_image_url = models.URLField()
    created_at = models.DateTimeField(auto_now_add=True)

```

Рис. 3.6 — Реалізація сутності *Новина* в додатку news

На рисунку 3.6 представлено реалізацію сутності *Новина* в додатку news.

На рисунку 3.7 показано реалізацію ComponentQueryMixin, тобто представлення списку компонентів з підтримкою сортування, пошуку та фільтрування:

```
class ComponentQueryMixin:
    allowed_sorts = [
        'name', '-name',
        'manufacturer', '-manufacturer',
        'operating_voltage', '-operating_voltage',
        'operating_current', '-operating_current',
        'power', '-power',
        'package_type', '-package_type',
    ]

    def get_filtered_sorted_queryset(self, base_queryset):
        request = self.request
        sort = request.GET.get('sort')

        query = request.GET.get('q')
        if query:
            base_queryset = base_queryset.filter(
                Q(name__icontains=query) |
                Q(manufacturer__icontains=query) |
                Q(description__icontains=query) |
                Q(category__name__icontains=query)
            )

        category = request.GET.get('category')
        if category:
            base_queryset = base_queryset.filter(category__name=category)

        package_type = request.GET.get('package_type')
        if package_type:
            base_queryset = base_queryset.filter(package_type=package_type)

        vmin, vmax = request.GET.get('voltage_min'),
            request.GET.get('voltage_max')
        base_queryset = base_queryset.filter(operating_voltage__gte=vmin)
        if vmax:
            base_queryset = base_queryset.filter(operating_voltage__lte=vmax)

        cmin, cmax = request.GET.get('current_min'),
            request.GET.get('current_max')
        base_queryset = base_queryset.filter(operating_current__gte=cmin)
        if cmax:
            base_queryset = base_queryset.filter(operating_current__lte=cmax)

        pmin, pmax = request.GET.get('power_min'), request.GET.get('power_max')
        if pmin:
            base_queryset = base_queryset.filter(power__gte=pmin)
        if pmax:
            base_queryset = base_queryset.filter(power__lte=pmax)

        if sort in self.allowed_sorts:
            base_queryset = base_queryset.order_by(sort)

        return base_queryset
```

Рис. 3.7 — Реалізація представлення списку компонентів

На рисунку 3.8 показано реалізацію функціоналу, необхідного для списку улюблених компонентів (додавання та видалення компоненту, перегляд списку):

```
@login_required
def add_to_favorites(request, pk):
    component = get_object_or_404(Component, pk=pk)
    request.user.favorite_components.add(component)
    messages.success(request, f"{component.name} додано до улюблених.")
    return redirect(request.META.get('HTTP_REFERER', 'component-list'))

@login_required
def remove_from_favorites(request, pk):
    component = get_object_or_404(Component, pk=pk)
    request.user.favorite_components.remove(component)
    messages.success(request, f"{component.name} видалено з улюблених.")
    return redirect(request.META.get('HTTP_REFERER', 'component-list'))

class FavoriteComponentListView(ComponentListView):
    def get_queryset(self):
        base_qs = super().get_queryset()
        return base_qs.filter(favorited by=self.request.user)
```

Рис. 3.8 — Реалізація функціоналу для списку улюблених компонентів

На рисунку 3.9 показано реалізацію форми для створення та редагування електронних компонентів:

```
class ComponentForm(forms.ModelForm):
    class Meta:
        model = Component
        fields = [
            'name', 'manufacturer', 'category', 'description',
            'package_type', 'operating_voltage', 'operating_current', 'power',
            'image_url'
        ]
        labels = {
            'name': 'Назва',
            'manufacturer': 'Виробник',
            'category': 'Категорія',
            'description': 'Опис',
            'package_type': 'Тип корпусу',
            'operating_voltage': 'Робоча напруга (V)',
            'operating_current': 'Робочий струм (A)',
            'power': 'Потужність (W)',
            'image_url': 'Зображення (URL)',
        }

        widgets = {
            'image_url': forms.URLInput(attrs={'placeholder': 'https://...'}),
        }
```

Рис. 3.9 — Реалізація форми ComponentForm

На рисунку 3.10 показано реалізацію форми, яка дозволяє авторизованим користувачам надсилати скарги на компоненти, якщо інформація про них неповна чи некоректна:

```
class ReportForm(forms.ModelForm):
    class Meta:
        model = Report
        fields = ['comment']

    comment = forms.CharField(
        label='Будь ласка, опишіть проблему з даним компонентом:',
        widget=forms.Textarea(attrs={
            'placeholder': 'Ваш коментар...',
            'rows': 10,
            'cols': 150
        })
    )
```

Рис. 3.10 — Реалізація форми ReportForm

На рисунку 3.11 показано реалізацію форми, що дозволяє адміністраторам створювати та редагувати категорії:

```
class CategoryForm(forms.ModelForm):
    class Meta:
        model = Category
        fields = ['name', 'parent', 'description']
        labels = {
            'name': 'Назва категорії',
            'parent': 'Батьківська категорія',
            'description': 'Опис (необов'язково)',
        }
```

Рис. 3.11 — Реалізація форми CategoryForm

На рисунку 3.12 показана реалізація функції, яка знаходить новини зі сфери електроніки на зовнішніх ресурсах та зберігає їх скорочені версії до бази даних:

```

def scrape_news(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)' # pretend to be a browser
    }
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.text, 'html.parser')

    for article in soup.select('div.post-single'):
        title_tag = article.select_one('h2.post-title a')
        content_tag = article.select_one('div.post-content')
        image_tag = article.find('img')

        if not title_tag:
            continue

        title = title_tag.text.strip()
        link = title_tag['href']
        cover_image_url = image_tag['src'] if image_tag else ''

        # Get first paragraph inside the content div (if exists)
        thumbnail = ''
        if content_tag:
            first_p = content_tag.find('p')
            if first_p:
                thumbnail = first_p.text.strip()

        NewsItem.objects.create(
            title=title,
            link=link,
            thumbnail=thumbnail,
            cover_image_url=cover_image_url,
        )

```

Рис. 3.12 — Реалізація функції `scrape_news`

Для отримання новин у проєкті використано технологію вебскрейпінгу – автоматизованого процесу зчитування та обробки HTML-даних з вебсторінок. Такий підхід дозволяє інтегрувати новинний контент із зовнішніх джерел без потреби у спеціальних API. Вебскрейпінг особливо корисний тоді, коли сторонній ресурс не надає прямого програмного доступу до даних, але має структуровану HTML-розмітку, яку можна аналізувати.

У реалізації вебскрейпінгу застосовано бібліотеку BeautifulSoup – популярний інструмент для розбору HTML-документів у Python. Вона дозволяє перетворити HTML-код сторінки у зручну для навігації структуру (дерево тегів), завдяки чому можна легко знаходити необхідні елементи за допомогою CSS-селекторів або назв тегів. BeautifulSoup підтримує різні парсери (наприклад, `html.parser`, `lxml`) і добре справляється з неідеально

розміченим HTML.

У файлі `news/scrapper.py` функція `scrape_news` надсилає HTTP-запит до сторінки з новинами, маскуючись під звичайний веббраузер за допомогою заголовка `User-Agent`. Отриманий HTML-код обробляється через `BeautifulSoup`, після чого за допомогою CSS-селекторів з нього витягуються окремі статті (елементи з класом `post-single`). Із кожної такої статті береться заголовок, посилання, перше зображення (як обкладинка) та перший абзац тексту (як короткий опис). Уся зібрана інформація зберігається в базу даних через Django-модель `NewsItem`.

На рисунку 3.13 показано реалізацію реєстрації, входу та виходу з облікового запису. Для цього використано вбудовані в Django класи `UserCreationForm`, `LoginView`, `LogoutView` із `django.contrib.auth`.

```
class RegisterView(CreateView):
    form_class = UserCreationForm
    template_name = 'users/register.html'
    success_url = reverse_lazy('component-list')

    def form_valid(self, form):
        user = form.save()
        login(self.request, user)
        return redirect(self.success_url)

class UserLoginView(LoginView):
    template_name = 'users/login.html'

class UserLogoutView(LogoutView):
    template_name = 'users/logout.html'
```

Рис. 3.13 — Фрагмент файла `users/views.py`

Для представлення вебдодатку користувачам (фронтенд-частина) використано HTML-шаблони вебсторінок проєкту. Наприклад, на рисунку 3.14 показано перелік шаблонів, необхідних для роботи каталогу. Для створення більш привабливого інтерфейсу було використано Bootstrap — популярний CSS-фреймворк, який містить готові стилі та компоненти для швидкої розробки веб-інтерфейсів.

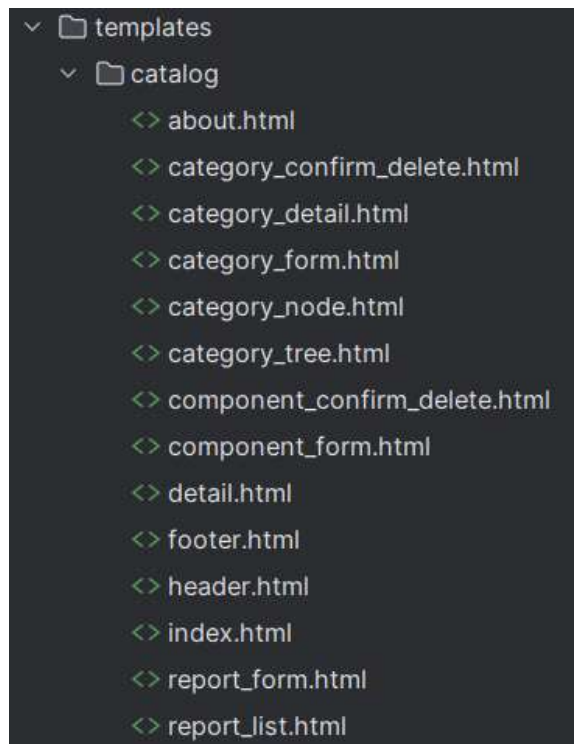


Рис. 3.14 — Перелік HTML-шаблонів для Django-додатку catalog

3.5. Подальші плани щодо розвитку продукту

Інформаційна система для пошуку електронних компонентів на даному етапі розробки вже може виконувати свої основні, а також деякі додаткові функції. Проте, для залучення більшої кількості користувачів та просування продукту варто продовжити розвивати проєкт.

Зокрема, користувачам було б цікаво знати про ціни та наявність електронних компонентів у різних продавців. Тому варто розробити механізм отримання цієї інформації (наприклад, через зовнішні API або з допомогою вебскрейпінгу) та відображення її на сторінках вебсайту.

Крім того, доцільно реалізувати систему порівняння цін і умов доставки від різних постачальників, що дозволить користувачам приймати більш обґрунтовані рішення. Також варто додати функцію сповіщення про зміни цін або появу товару в наявності – це підвищить зручність користування сайтом та заохотить користувачів повертатися. Подальший розвиток може включати підтримку багатьох мов, інтеграцію з сервісами авторизації через Google або інші системи, а також створення мобільної

версії інтерфейсу або окремого застосунку. Усе це зробить систему більш гнучкою, конкурентоспроможною та орієнтованою на кінцевого користувача.

Висновки до розділу 3

У третьому розділі було здійснено проєктування та реалізацію інформаційної системи для пошуку електронних компонентів, що охоплює всі етапи – від моделювання структури даних до створення функціонального вебзастосунку.

Початковим етапом стало створення ER-моделі (моделі “сутність-зв’язок”), яка дозволила чітко описати ключові об’єкти предметної області (компоненти, категорії, виробники, постачальники, користувачі) та їхні взаємозв’язки. Це забезпечило логічну основу для формування цілісної та узгодженої структури даних. На основі ER-моделі було здійснено проєктування реляційної бази даних, оптимізованої під основні запити користувачів. Особливу увагу приділено забезпеченню нормалізації та можливості розширення структури бази даних без втрати продуктивності.

Наступним кроком стало визначення структури проєкту на базі фреймворку Django. Обрано архітектурний поділ за принципом “apps” для модульності – кожен функціональний блок (наприклад, компоненти, пошук, користувачі) ізольовано в окремому застосунку, що спрощує підтримку та тестування. Безпосередня розробка системи охоплювала реалізацію інтерфейсів взаємодії з базою даних, API для пошуку й фільтрації компонентів, механізмів автентифікації користувачів, а також шаблонів відображення даних у вебінтерфейсі. Особливу увагу було приділено юзабіліті та швидкодії, а також підтримці багатомовності.

Було сформульовано плани подальшого розвитку, серед яких – інтеграція зі сторонніми API постачальників (Digi-Key, Octopart), розширення фільтрації за параметрами (напруга, частота, розміри), мобільна версія системи та підтримка AI-рекомендацій на основі історії пошуків користувача.

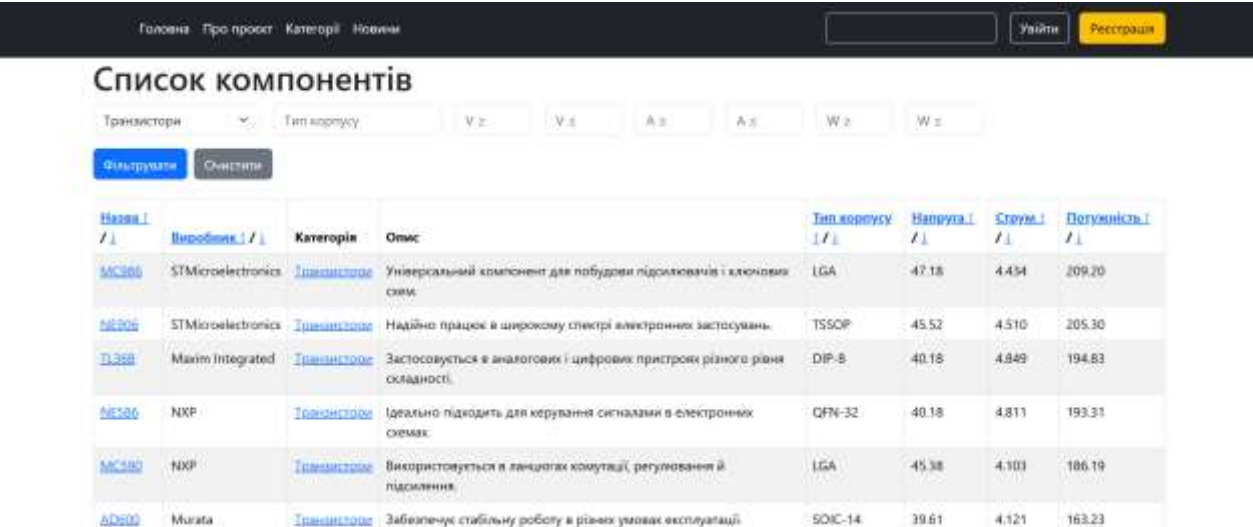
						56

4. ТЕСТУВАННЯ РОЗРОБКИ

4.1. Тестування основного функціоналу

Для тестування розробки до бази даних було додано автоматично згенеровані тестові компоненти (по 50 об'єктів у кожній категорії).

На головній сторінці, яка відкривається незареєстрованим користувачем, доступне зручне меню, а також каталог із основними функціями – пошук, фільтрування, сортування за різними параметрами. На рисунку 4.1 показано частину головної сторінки із використанням сортування за потужністю (за спаданням) та фільтром за категорією «Транзистори».

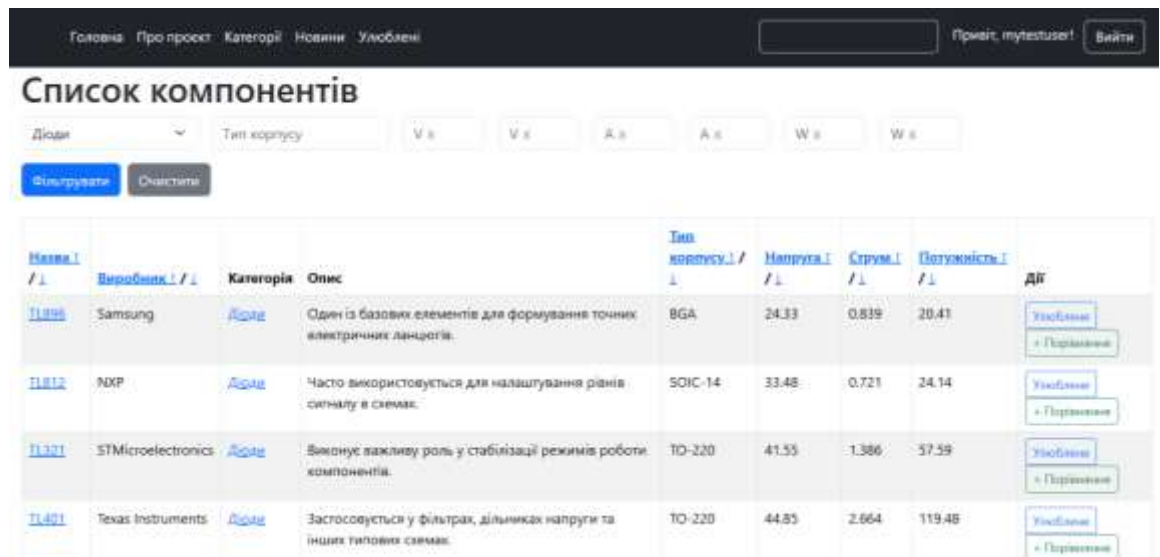


The screenshot shows a web interface for a component catalog. At the top, there is a navigation bar with links: 'Головна', 'Про проект', 'Категорії', 'Новини'. On the right, there are buttons for 'Увійти' and 'Регистрація'. Below the navigation bar, the title 'Список компонентів' is displayed. Underneath, there are filters for 'Транзистори' and 'Тип корпусу', and sorting options: 'V з', 'V з', 'A з', 'A з', 'W з', 'W з'. There are also buttons for 'Фільтрувати' and 'Очистити'. The main content is a table with columns: 'Назва', 'Виробник', 'Категорія', 'Опис', 'Тип корпусу', 'Напруга', 'Струм', and 'Потужність'. The table lists several transistors from manufacturers like STMicroelectronics, Maxim Integrated, NXP, and Murata.

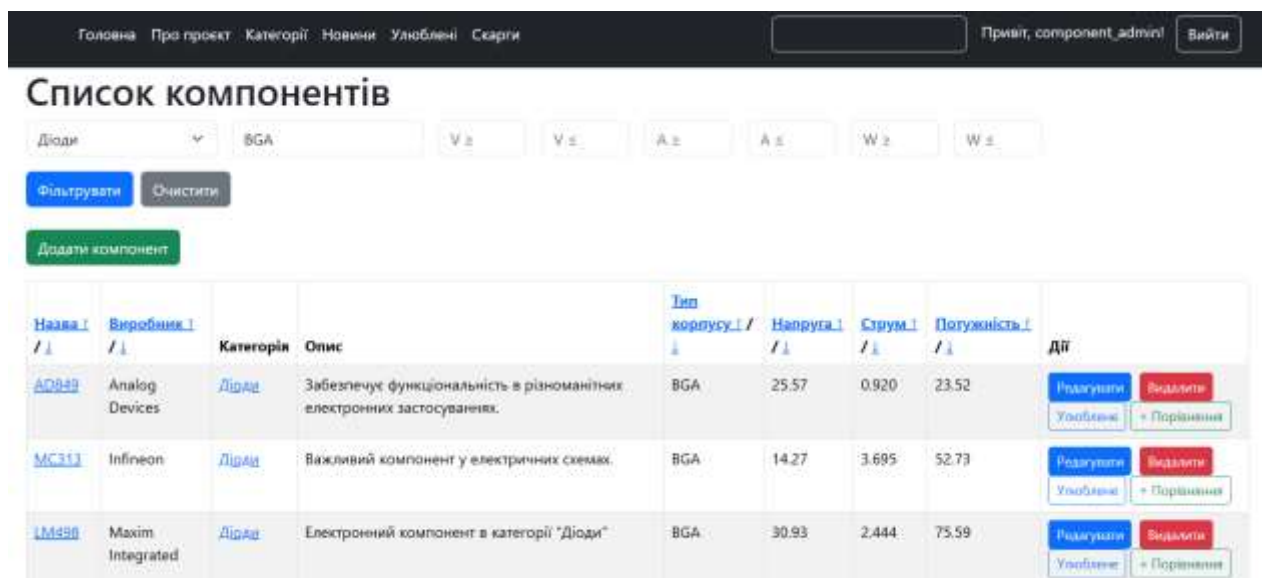
Назва	Виробник	Категорія	Опис	Тип корпусу	Напруга	Струм	Потужність
MC385	STMicroelectronics	Транзистори	Універсальний компонент для побудови підсилювачів і ключових схем.	LGA	42.18	4.434	209.20
ME205	STMicroelectronics	Транзистори	Надійно працює в широкому спектрі електронних застосувань.	TSSOP	45.52	4.510	205.30
TL358	Maxim Integrated	Транзистори	Застосовується в аналогових і цифрових пристроях різного рівня складності.	DIP-8	40.18	4.849	194.83
ME286	NXP	Транзистори	Ідеально підходить для керування сигналами в електронних схемах.	QFN-32	40.18	4.811	193.31
MC385	NXP	Транзистори	Використовується в ланцюгах комутації, регулювання й підсилення.	LGA	45.38	4.103	186.19
AO500	Murata	Транзистори	Забезпечує стабільну роботу в різних умовах експлуатації.	SOIC-14	39.61	4.121	163.23

Рис. 4.1 — Головна сторінка для гостя сайту із використанням сортування та фільтрів

Зареєстрованому користувачеві, який не є адміністратором, доступні всі вищеперераховані функції, а також є можливість додавати (чи видаляти) компоненти до списку улюблених, та додавати чи видаляти компоненти для порівняльної таблиці. На рисунку 4.2 показано частину головної сторінки для зареєстрованого користувача після пошуку за назвою компонента.



Для адміністраторів додається можливість редагувати, додавати чи видаляти компоненти. На рисунку 4.3 показано частину головної сторінки для адміністратора із використаними фільтрами за категорією та типом корпусу.



AD849



Виробник: Analog Devices

Категорія: Діоди

Опис: Забезпечує функціональність в різноманітних електронних застосуваннях.

Технічні характеристики

- Тип корпусу: SGA
- Робоча напруга (V): 25.57
- Робочий струм (A): 0.920
- Потужність (W): 23.52

Рис. 4.4 — Приклад сторінки з інформацією про окремий компонент, яка доступна гостям сайту

Зареєстрований користувач також має можливість подати скаргу, якщо інформація про компонент є неповною чи некоректною, а також додати до списку улюблених, як показано на рисунку 4.5.

AD849



Виробник: Analog Devices

Категорія: Діоди

Опис: Забезпечує функціональність в різноманітних електронних застосуваннях.

[Улюблені](#)

Технічні характеристики

- Тип корпусу: SGA
- Робоча напруга (V): 25.57
- Робочий струм (A): 0.920
- Потужність (W): 23.52

[Відсутня чи некоректна інформація? Надішліть скаргу](#)

Рис. 4.5 — Приклад сторінки з інформацією про окремий компонент для зареєстрованого користувача

При перегляді списку категорій адміністратор може додавати нові категорії, а також змінювати чи видаляти наявні (рис. 4.6), на відміну від інших користувачів, які можуть лише переглядати список (рис 4.7).

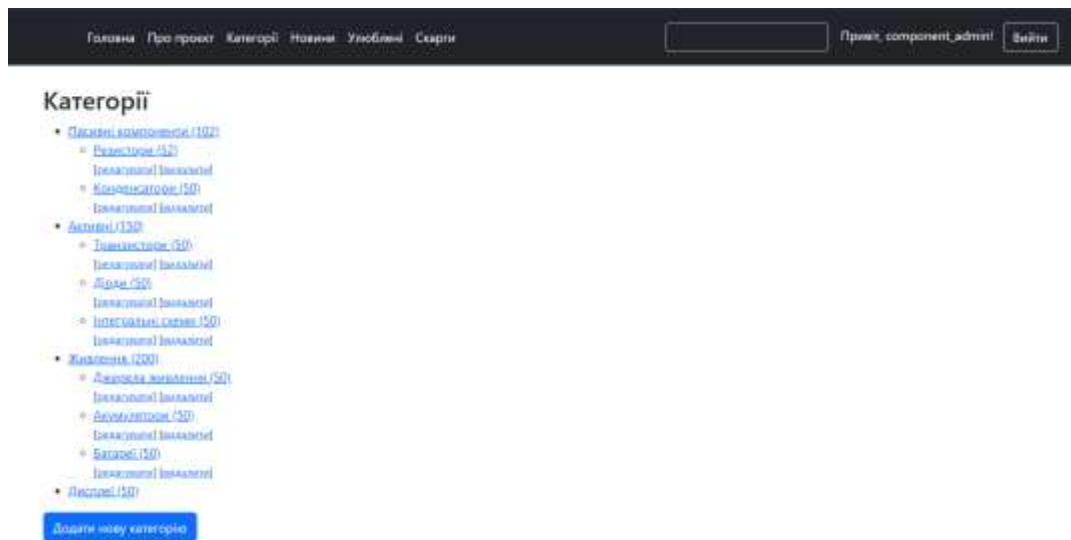


Рис. 4.6 — Список категорій для адміністратора

Категорії

- [Пасивні компоненти \(102\)](#)
 - [Резистори \(52\)](#)
 - [Конденсатори \(50\)](#)
- [Активні \(150\)](#)
 - [Транзистори \(50\)](#)
 - [Діоди \(50\)](#)
 - [Інтегральні схеми \(50\)](#)
- [Живлення \(200\)](#)
 - [Джерела живлення \(50\)](#)
 - [Акумулятори \(50\)](#)
 - [Батареї \(50\)](#)
- [Дисплеї \(50\)](#)

Рис. 4.7 — Список категорій, доступний гостям та зареєстрованим користувачам

4.2. Тестування додаткового функціоналу

Будь-який зареєстрований користувач може створити для себе список улюблених компонентів для більш зручного доступу. На рисунку 4.8 показано приклад такого списку.

Назва	Виробник	Категорія	Опис	Тип корпусу	Напруга	Струм	Потужність	Дії
AD640	Analog Devices	Діоди	Забезпечує функціональність в різноманітних електронних застосуваннях.	BGA	25.57	0.920	23.52	Видати з улюблених + Порівняти
CRCW060310K0FKEA	Vishay	Резистори	Компактний SMD резистор серії CRCW з опором 10 кОм, точністю ±1%, стандартним розміром 0603, підход...	—	—	—	1.00	Видати з улюблених + Порівняти

Рис. 4.8 — Приклад списку улюблених компонентів

Також кожен зареєстрований користувач має право подати скаргу на деякий електронний компонент, якщо інформація недійсна чи неповна. На рисунку 4.9 показано форму для такої скарги.

Відправити скаргу на об'єкт CRCW060310K0FKEA

Будь ласка, опишіть проблему з даним компонентом:

Не вказано тип корпусу

[Відправити скаргу](#)

[Назад](#)

Рис. 4.9 — Приклад форми для заповнення скарги на компонент

Кожен адміністратор сайту може подивитися список скарг та позначати окрему скаргу як «оброблену» чи «в обробці». Можна подивитися інформацію про скаргу чи перейти за посиланням на компонент, щоб перевірити дані. На рисунку 4.10 показано сторінку зі списком скарг.

Скарги

Список скарг від користувачів щодо неповних чи некоректних даних компонентів

Компонент	Скарга	Дата відправлення	Статус	Дія
MC216	Потужність вказана неправильно	May 28, 2025, 1:43 a.m.	В обробці	Позначити як оброблено
TL490	Неправильно вказана категорія	May 28, 2025, 1:41 a.m.	В обробці	Позначити як оброблено
TL490	Зображення не відповідає дійсності	May 28, 2025, 1:39 a.m.	В обробці	Позначити як оброблено
AD849	Недостатньо детальний опис	May 28, 2025, 1:39 a.m.	В обробці	Позначити як оброблено
CRCW060310K0FKEA	Не вказано тип корпусу	May 28, 2025, 1:37 a.m.	Оброблено	Позначити як в обробці

Рис. 4.10 — Список скарг, доступний лише адміністраторам

Щоб обробити скаргу, адміністратор може змінити інформацію про компонент. З допомогою кнопки «Редагувати» можна перейти до форми редагування компонента, яка показана на рисунку 4.11. Також адміністратори мають право видаляти компоненти або додавати нові.

Редагувати компонент

Назва

Виробник

Категорія

Опис

Компактний SMD резистор серії CRCW з опором 10 кОм, точністю ±1%, стандартним розміром 0603, підходить для широкого спектру застосувань у цифровій і аналоговій електроніці.

Тип корпусу

Робоча напруга (V)

Робочий струм (A)

Потужність (W)

Зображення (URL)

Зберегти

Скасувати

Рис. 4.11 — Форма для редагування електронних компонентів

Аналогічно, адміністратори можуть змінювати, видаляти чи додавати категорії. На рисунку 4.12 показано форму для редагування категорії, яка дозволяє змінити назву, батьківську категорію (якщо така є), та опис, що буде відображено на сторінці категорії.

Редагувати категорію

Назва категорії

Батьківська категорія

Опис (необов'язково)

Конденсатор є пасивним електронним компонентом і широко застосовується в електронних схемах для блокування постійного струму, пропускаючи змінний струм.

Зберегти

Рис. 4.12 — Приклад форми для редагування категорії

При перегляді списку компонентів, зареєстрований користувач може додавати деякі з них до списку для порівняння. Після додавання хоча б двох компонентів, користувач може натиснути кнопку «Перейти до порівняння» та переглянути порівняльну таблицю.

Таке представлення дозволяє зручніше обирати, який саме із подібних компонентів потрібен користувачу. Приклад такої таблиці показано на рисунку 4.13.

Порівняння компонентів

Атрибут	MC986	NE906	TL368
Зображення			
Назва	MC986	NE906	TL368
Виробник	STMicroelectronics	STMicroelectronics	Maxim Integrated
Категорія	Транзистори	Транзистори	Транзистори
Опис	Універсальний компонент для побудови підсилювачів і ключових схем.	Надійно працює в широкому спектрі електронних застосувань.	Застосовується в аналогових і цифрових пристроях різного рівня складності.
Тип корпусу	LGA	TSSOP	DIP-8
Робоча напруга (V)	47.18	45.52	40.18
Робочий струм (A)	4.434	4.510	4.849
Потужність (W)	209.20	205.30	194.83

Рис. 4.13 — Приклад порівняльної таблиці для трьох компонентів

4.3. Тестування новинного функціоналу

Користувачів даної інформаційної системи, ймовірно, цікавлять останні новини зі світу електроніки. Тому в якості додаткового функціоналу було реалізовано окрему сторінку, яка агрегує новини з авторитетних джерел у сфері електроніки.

Функціонал реалізовано за допомогою вебскрейпінгу, що дозволяє автоматично збирати найсвіжіші статті з обраних сайтів. Кожна новина відображається із заголовком, коротким описом, датою публікації та активним посиланням на першоджерело. На рисунку 4.14 продемонстровано приклад інтерфейсу цієї сторінки.

У ході тестування було перевірено:

- стабільність завантаження новин з кількох джерел;
- обробку ситуацій, коли новини тимчасово недоступні;
- правильність парсингу структури HTML-сторінок;
- відповідність відображення даних у фронтенді;
- коректність роботи посилань на оригінальні ресурси.

						64

Тестування підтвердило, що новинний модуль функціонує згідно з вимогами: інформація оновлюється регулярно, відображення в браузері стабільне, а навігація до джерел працює без помилок.

[Головна](#) [Про проект](#) [Категорії](#) [Новини](#) [Улюблені](#)

[Привіт, mytester!](#) [Вийти](#)

Рис. 4.14 — Сторінка із актуальними новинами

Висновки до розділу 4

Розділ присвячений перевірці якості програмної реалізації інформаційної системи через поетапне тестування її функціоналу.

Тестування охопило як основні, так і додаткові компоненти системи, а також новинний модуль, що забезпечує користувачів актуальною інформацією.

Проведено тестування основного функціоналу, зокрема — реєстрації, входу в систему, пошуку та фільтрації електронних компонентів, перегляду детальної інформації про товар, а також взаємодії з базою даних. Перевірки підтвердили коректну роботу бізнес-логіки, відсутність критичних помилок і стабільність системи при різних сценаріях використання.

Розглянуто додатковий функціонал, зокрема: створення облікового запису постачальника, редагування профілю, додавання нових компонентів до бази, керування замовленнями та застосування ролей користувачів. Результати показали, що реалізований функціонал працює відповідно до вимог і забезпечує належний рівень безпеки доступу.

Новинний функціонал передбачає відображення актуальних новин, пов'язаних із ринком електронних компонентів, із можливістю автоматичного оновлення стрічки новин через API або ручного введення. Тестування показало, що інтерфейс новин інтегрується з основною системою без конфліктів, а інформація відображається динамічно та своєчасно.

Загалом, розроблена система пройшла комплексне функціональне тестування. Виявлені під час перевірок недоліки були оперативно усунені.

						66

ВИСНОВКИ

У межах даного проєкту було реалізовано повнофункціональну інформаційну систему для пошуку електронних компонентів, яка дозволяє зручно й ефективно працювати з великим обсягом технічної інформації. Основною метою роботи було створення інструменту, який спростить доступ до структурованих даних про електронні компоненти для інженерів, технічних фахівців, студентів та всіх, хто займається розробкою електроніки. Це завдання було успішно виконано: розроблена система забезпечує зручний інтерфейс, можливість пошуку, перегляду й аналізу даних за різними критеріями.

У процесі дослідження було вивчено принципи побудови інформаційних систем, проаналізовано предметну область, визначено вимоги користувачів і проведено порівняння з існуючими аналогами. Це дозволило чітко сформулювати функціональні вимоги та визначити оптимальні технології для реалізації проєкту. Була обрана сучасна архітектура, що передбачає використання надійної СУБД, популярного фреймворку та мови програмування, що забезпечують масштабованість і можливість подальшого розвитку системи.

Розробка проходила у кілька етапів – від проєктування бази даних до реалізації основного функціоналу та тестування. Система успішно пройшла перевірку на тестових даних: усі функції працюють коректно, відповідаючи поставленим вимогам. Було перевірено її роботу при різних сценаріях використання та з різними типами запитів.

Загалом, проєкт продемонстрував, що створення спеціалізованих інформаційних систем для вузьких технічних напрямів є не лише актуальним, а й практично необхідним. У світі, де обсяги технічної інформації невпинно зростають, ефективне управління цими даними стає критичним фактором для продуктивної інженерної діяльності.

Результати проєкту можуть бути використані як база для подальших

						67

розробок: розширення функціоналу, додавання фільтрів, інтеграція з зовнішніми базами, підтримка експорту даних тощо. Запропонована система має потенціал для використання як у навчальному процесі, так і в реальних інженерних завданнях, слугуючи зручним інструментом для пошуку, аналізу та вибору електронних компонентів.

Таким чином, дане дослідження підтвердило доцільність розробки спеціалізованої інформаційної системи в галузі електроніки та продемонструвало її ефективність у вирішенні прикладних завдань пошуку та структуризації технічної інформації.

						68

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Britannica // [Електроний ресурс]. — Режим доступу: <https://www.britannica.com/topic/information-system> — Дата доступу: березень 2025
2. “Clean Architecture: A Craftsman's Guide to Software Structure and Design” by Robert C. Martin
3. DB-Engines Ranking // [Електроний ресурс]. — Режим доступу: <https://db-engines.com/en/ranking> — Дата доступу: квітень 2025
4. “Design Patterns: Elements of Reusable Object-Oriented Software” by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
5. DigiKey // [Електроний ресурс]. — Режим доступу: <https://www.digikey.com/> — Дата доступу: березень 2025
6. Django Documentation // [Електроний ресурс]. — Режим доступу: <https://docs.djangoproject.com/en/5.2/> — Дата доступу: квітень 2025
7. Electronic Components // [Електроний ресурс]. — Режим доступу: <https://www.geeksforgeeks.org/electronic-components/> — Дата доступу: квітень 2025
8. Entity Relation Model // [Електроний ресурс]. — Режим доступу: <https://www.geeksforgeeks.org/introduction-of-er-model/> — Дата доступу: квітень 2025
9. FastAPI Documentation // [Електроний ресурс]. — Режим доступу: <https://fastapi.tiangolo.com/> — Дата доступу: квітень 2025
10. Flask Documentation // [Електроний ресурс]. — Режим доступу: <https://flask.palletsprojects.com/en/stable/> — Дата доступу: квітень 2025
11. Mouser Electronics // [Електроний ресурс]. — Режим доступу: <https://eu.mouser.com/> — Дата доступу: березень 2025
12. MySQL Documentation // [Електроний ресурс]. — Режим доступу: <https://dev.mysql.com/doc/> — Дата доступу: квітень 2025
13. Octopart // [Електроний ресурс]. — Режим доступу: <https://octopart.com/>

						69

— Дата доступу: березень 2025

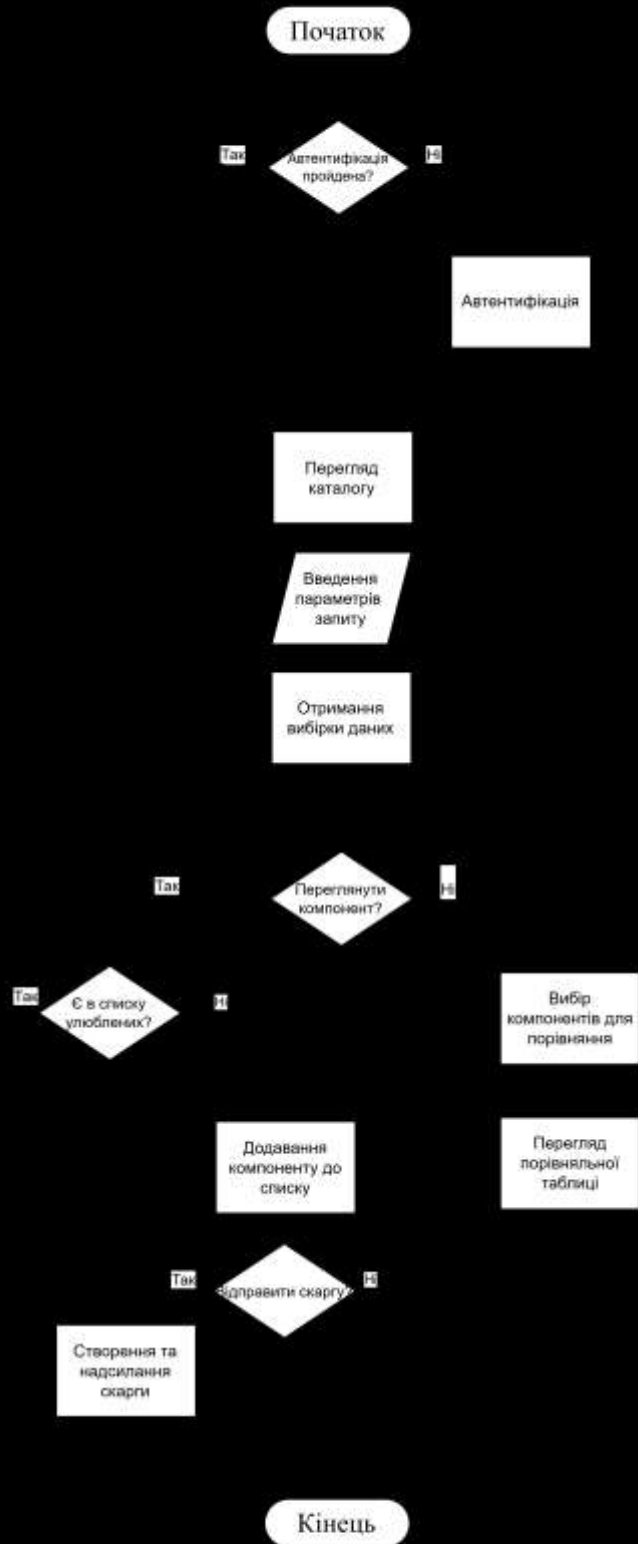
14. PostgreSQL Documentation // [Електроний ресурс]. — Режим доступу: <https://www.postgresql.org/docs/> — Дата доступу: квітень 2025
15. Python Documentation // [Електроний ресурс]. — Режим доступу: <https://docs.python.org/3/> — Дата доступу: квітень 2025
16. Python introduction // [Електроний ресурс]. — Режим доступу: https://www.w3schools.com/python/python_intro.asp — Дата доступу: квітень 2025
17. “Refactoring: Improving the Design of Existing Code” by Martin Fowler
18. SQLite Documentation // [Електроний ресурс]. — Режим доступу: <https://www.sqlite.org/> — Дата доступу: квітень 2025
19. What is a database schema // [Електроний ресурс]. — Режим доступу: <https://www.ibm.com/think/topics/database-schema> — Дата доступу: квітень 2025
20. What is NoSQL // [Електроний ресурс]. — Режим доступу: <https://www.oracle.com/ua/database/nosql/what-is-nosql/> — Дата доступу: квітень 2025
21. What is SQL // [Електроний ресурс]. — Режим доступу: <https://aws.amazon.com/what-is/sql/> — Дата доступу: квітень 2025

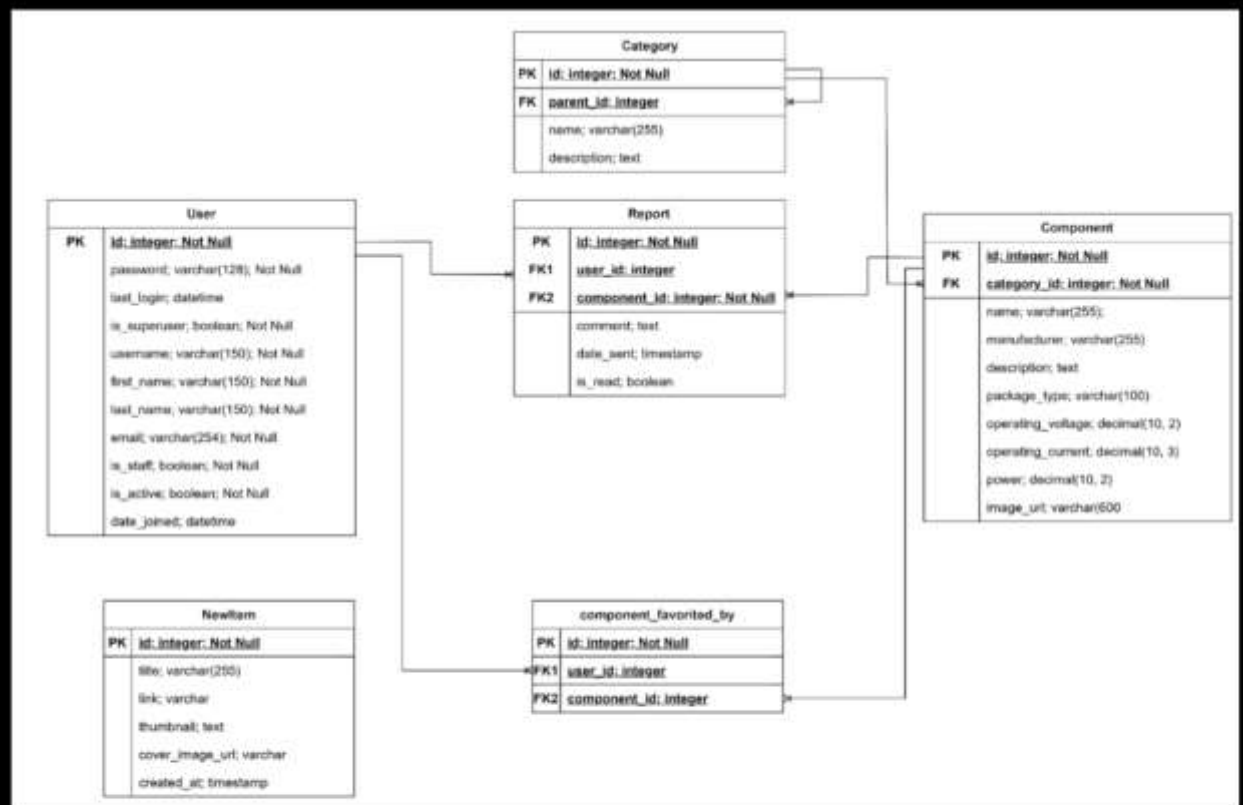
ДОДАТКИ

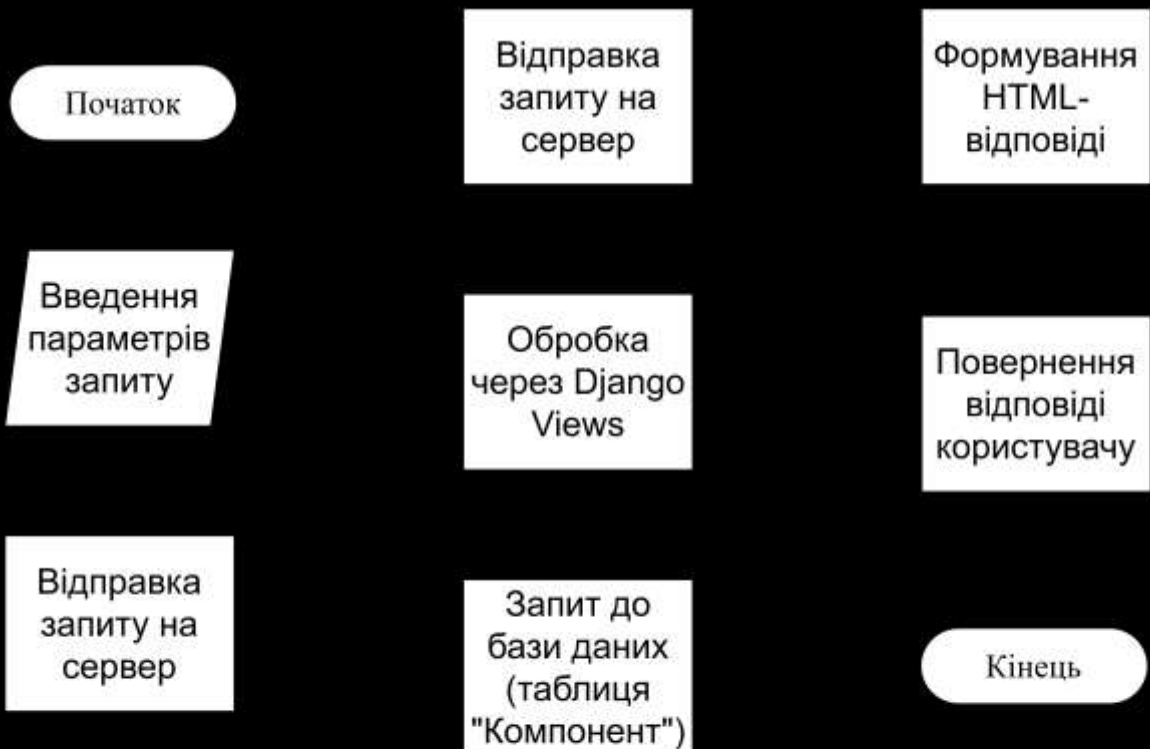
ДОДАТОК А
Інформаційна система пошуку електронних компонентів
Копії графічних матеріалів

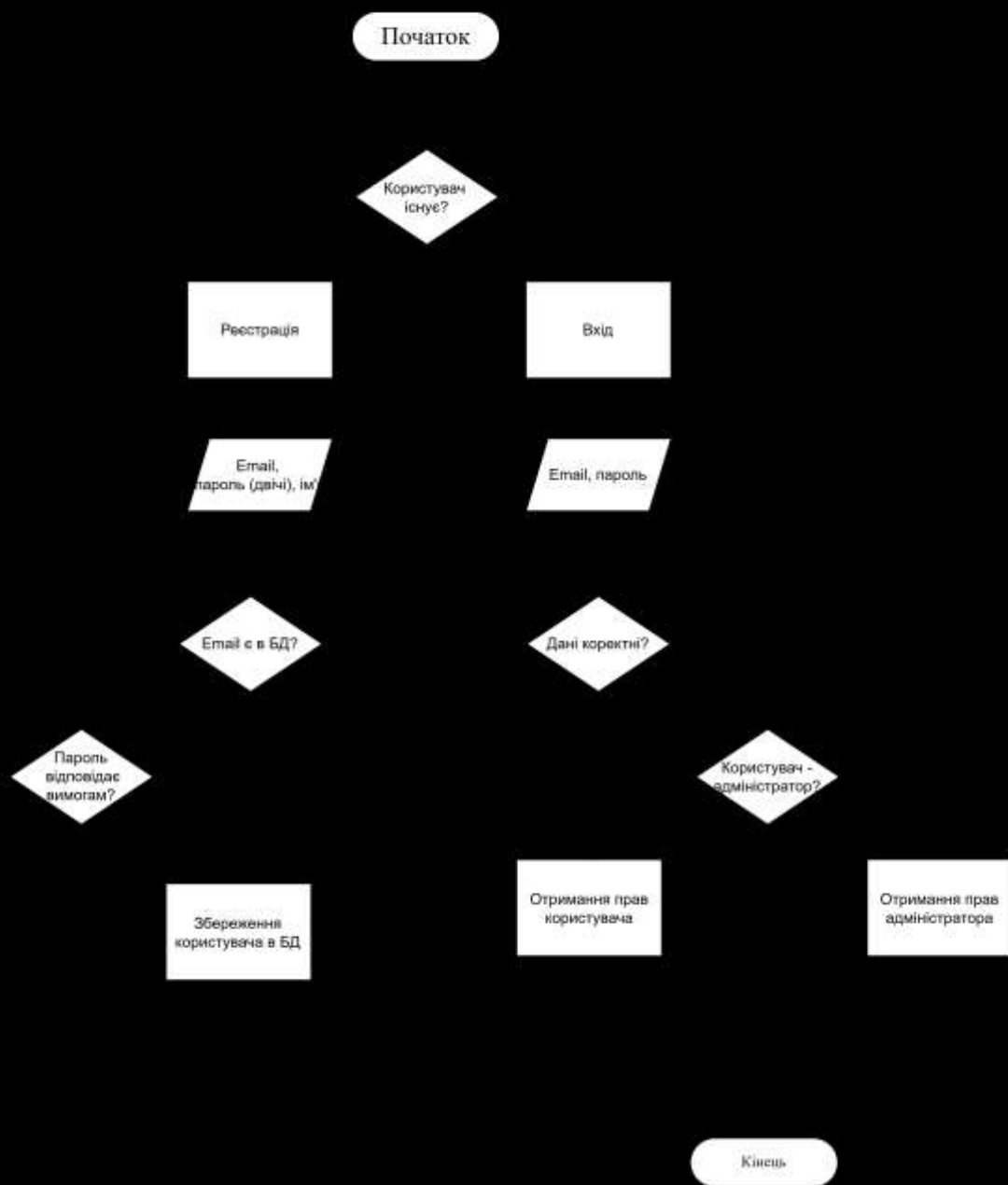
Аркушів 4

Київ – 2025 р.









ДОДАТОК Б
Інформаційна система пошуку електронних компонентів
Лістинг програми

Аркушів 1

Київ – 2025 р.

catalog/models.py

```
class Component(models.Model):
    name = models.CharField(max_length=255)
    manufacturer = models.CharField(max_length=255)
    category = models.ForeignKey('Category', on_delete=models.CASCADE)
    description = models.TextField(null=True, blank=True)
    favorited_by = models.ManyToManyField(User, related_name='favorite_components', blank=True)

    package_type = models.CharField("Тип корпусу", max_length=100, null=True, blank=True)
    operating_voltage = models.DecimalField("Робоча напруга (V)", max_digits=10, decimal_places=2, null=True,
                                           blank=True)
    operating_current = models.DecimalField("Робочий струм (A)", max_digits=10, decimal_places=3, null=True, blank=True)
    power = models.DecimalField("Потужність (W)", max_digits=10, decimal_places=2, null=True, blank=True)

    image_url = models.URLField("Зображення (URL)", max_length=600, null=True, blank=True)

    def __str__(self):
        return f'{self.manufacturer} {self.name}'

    @property
    def display_image_url(self):
        DEFAULT_IMAGE_URL = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ4ks8qNKu-
        FxA4GcthRL2dkkKQxJB_q7dprgWvbGthnU6nn6B7FdQDVdEkkTauXDubOLk&usqp=CAU"
        return self.image_url or DEFAULT_IMAGE_URL
```

class Category(models.Model):

```
name = models.CharField(max_length=255)
parent = models.ForeignKey('self', null=True, blank=True, on_delete=models.CASCADE)
description = models.TextField(null=True, blank=True)

def __str__(self):
    return self.name

@property
def total_component_count(self):
    count = self.component_set.count()
    for child in self.category_set.all():
        count += child.total_component_count
    return count
```

class Report(models.Model):

```
component = models.ForeignKey("Component", on_delete=models.CASCADE, related_name="reports")
user = models.ForeignKey(User, on_delete=models.CASCADE)
comment = models.TextField()
date_sent = models.DateTimeField(auto_now_add=True)
is_read = models.BooleanField(default=False)

def __str__(self):
    return f"Report on {self.component.name} by {self.user.username}"
```

news/models.py

```
class NewsItem(models.Model):
    title = models.CharField(max_length=255)
    link = models.URLField()
    thumbnail = models.TextField()
    cover_image_url = models.URLField()
    created_at = models.DateTimeField(auto_now_add=True)
```

urls.py

```
urlpatterns = [
    path("", include('catalog.urls')),
    path('admin/', admin.site.urls),
    path('users/', include('users.urls')),
    path('news/', include('news.urls')),
]
```

ДОДАТОК В
Інформаційна система пошуку електронних компонентів
Копія презентації

Аркушів 10

Київ – 2025 р.

Інформаційна система пошуку електронних компонентів

Виконала: студентка IV курсу, групи KB-12
Гнатюк С.В.
Керівник: ст. викладач каф. СПІСКС
Радченко К. О.

Актуальність теми

- На ринку електроніки представлено **десятки тисяч компонентів** з різними характеристиками та сумісністю.
- Пошук потрібного компонента без структурованої бази — **процес, що вимагає багато часу і ресурсу.**
- Інженери витрачають **багато часу на ручний пошук, порівняння та перевірку параметрів.**
- Відсутність єдиної системи ускладнює **швидке прийняття технічних рішень та уповільнює розробку.**
- Інформаційна система дозволяє **оптимізувати робочі процеси, зменшити ризики помилок та підвищити ефективність проєктування.**

Мета роботи:

- **Розробити інформаційну систему для пошуку електронних компонентів.**
 - **Забезпечити структурований, зручний доступ до технічних даних для інженерів та фахівців.**
-

Огляд наявних рішень

Для успішного виконання проєкту необхідно провести аналіз наявних на ринку рішень.

Було досліджено такі системи:

- Octopart
- DigiKey
- Mouser Electronics

The logo for Octopart, featuring the word "Octopart" in a blue sans-serif font, with a gear icon replacing the letter 'o'.

**MOUSER
ELECTRONICS**

Постановка задачі

- Розробити інформаційну систему для пошуку електронних компонентів у вигляді вебзастосунку з онлайн-каталогом
 - Впровадити зручну систему автентифікації
 - Забезпечити різні рівні доступу до функціоналу для гостей сайту, авторизованих користувачів та адміністраторів сайту
 - Надати можливість зручного перегляду каталогу із пошуком, сортуванням та фільтруванням даних
 - Додати функціонал для створення порівняльних таблиць та списків компонентів для зручного доступу
 - Забезпечити можливість вносити зміни до каталогу для адміністраторів
 - Додати систему зворотнього зв'язку для користувачів
 - Внести додатковий функціонал, який також може зацікавити користувачів системи
-

Обрані технології

Для розробки системи було обрано такі технології:

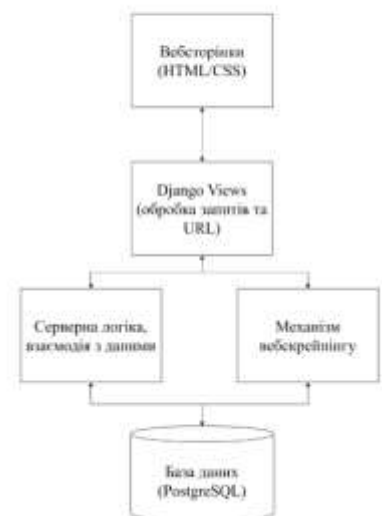
- Мова програмування - Python
- Фреймворк - Django
- Система управління базами даних - PostgreSQL



Структура системи

Загальна структура системи:

- Frontend для взаємодії з користувачами
- Backend для обробки запитів та відповідей
- База даних для зберігання інформації



Результати розробки

Було розроблено інформаційну систему у вигляді вебзастосунку.

Реалізовано такий функціонал, доступний всім відвідувачам сайту:

- Головна сторінка вебсайту зі зручним меню та повним списком електронних компонентів
- Перегляд списку категорій та компонентів у окремій категорії
- Сортування електронних компонентів каталогу за різними параметрами
- Пошук електронних компонентів за різними параметрами
- Фільтрування каталогу електронних компонентів
- Перегляд детальної інформації про кожен компонент
- Перегляд актуальних новин зі світу електроніки із зовнішніх ресурсів
- Реєстрація та вхід в обліковий запис

Результати розробки

Головна | Про компанію | Контакт | Реєстрація

Add

Remove

Список компонентів

Таблиця компонентів

—

По шуканню

[Обновить](#)

[Сбросить](#)

Имя / ID	Полное имя / ID	Категория	Вид	Таблицы ID	Имя ID	Свойства ID	Параметры ID
00000	Базовый элемент	Базовый элемент	Базовый элемент	0000	0000	0000	0000
00001	Базовый элемент	Базовый элемент	Базовый элемент	0001	0001	0001	0001
00002	Базовый элемент	Базовый элемент	Базовый элемент	0002	0002	0002	0002
00003	Базовый элемент	Базовый элемент	Базовый элемент	0003	0003	0003	0003
00004	Базовый элемент	Базовый элемент	Базовый элемент	0004	0004	0004	0004
00005	Базовый элемент	Базовый элемент	Базовый элемент	0005	0005	0005	0005
00006	Базовый элемент	Базовый элемент	Базовый элемент	0006	0006	0006	0006
00007	Базовый элемент	Базовый элемент	Базовый элемент	0007	0007	0007	0007
00008	Базовый элемент	Базовый элемент	Базовый элемент	0008	0008	0008	0008
00009	Базовый элемент	Базовый элемент	Базовый элемент	0009	0009	0009	0009

AD849



Производитель: Analog Devices

Категорія: Діади

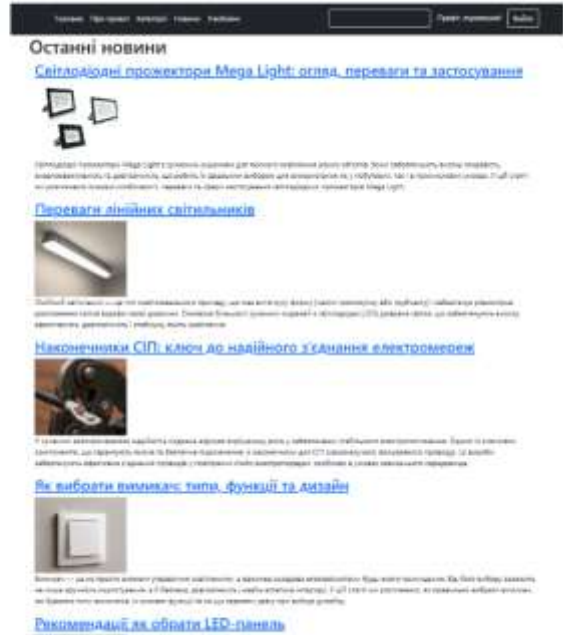
Опис: Забезпечує функціональність в різноманітних електронних застосуваннях.

Технічні характеристики

- Тип корпусу: BGA
- Робоча напруга (V): 25.57
- Робочий струм (A): 0.920
- Потужність (W): 23.52

Результати розробки Категорії

- Пасивні компоненти (102)
 - Резистори (52)
 - Конденсатори (50)
- Активні (150)
 - Транзистори (50)
 - Діоди (50)
 - Інтегральні схеми (50)
- Живлення (200)
 - Джерела живлення (50)
 - Акумулятори (50)
 - Батареї (50)
- Дисплеї (50)



Результати розробки

Для зареєстрованих користувачів також доступний такий функціонал:

- Можливість обрати кілька компонентів для створення порівняльної таблиці
- Створення та оновлення списку улюблених електронних компонентів
- Форма для надсилання скарги на електронний компонент, щоб повідомити про помилку
- Вихід з облікового запису

Назва 1 / 1	Виробник 1 / 1	Категорія	Опис	Тип корпусу 1 / 1	Напруга 1 / 1	Струм 1 / 1	Потужність 1 / 1	Дії
ACS92	Analog Devices	Діоди	Забезпечує функціональність в різноманітних електронних застосуваннях.	BGA	25.57	0.920	23.52	Видати з улюблених + Порівняти
CRCW060310K0FFEA	Vishay	Резистори	Компактний SMD резистор серії CRCW з опором 10 кОм, точністю ±1%, стандартним розміром 0603, підход...	—	—	—	1.00	Видати з улюблених + Порівняти

Результати розробки

Головна сторінка для зареєстрованого користувача

Форма для відправлення скарги на компонент

Головна | Про проект | Категорії | Новини | Контакти

Список компонентів

Виділяти | Додати

Назва	Виробник	Категорія	Опис	Дані	Додати до кошика	Додати до кошика	Додати до кошика	Додати до кошика
№	№	№	№	№	№	№	№	№
MC986	STMicroelectronics	Транзистори	Універсальний компонент для побудови підсилювачів і ключових схем.	209.20	47.18	4.434	209.20	Додати до кошика
NE906	STMicroelectronics	Транзистори	Надійно працює в широкому спектрі електронних застосувань.	205.30	45.52	4.510	205.30	Додати до кошика
TL368	Maxim Integrated	Транзистори	Застосовується в аналогових і цифрових пристроях різного рівня складності.	194.83	40.18	4.849	194.83	Додати до кошика

Відправити скаргу на об'єкт CRCW060310K0FKEA

Введіть текст скарги

Відправити скаргу

Результати розробки

Порівняння компонентів

Атрибут	MC986	NE906	TL368
Зображення			
Назва	MC986	NE906	TL368
Виробник	STMicroelectronics	STMicroelectronics	Maxim Integrated
Категорія	Транзистори	Транзистори	Транзистори
Опис	Універсальний компонент для побудови підсилювачів і ключових схем.	Надійно працює в широкому спектрі електронних застосувань.	Застосовується в аналогових і цифрових пристроях різного рівня складності.
Тип корпусу	LGA	TSSOP	DIP-8
Робоча напруга (V)	47.18	45.52	40.18
Робочий струм (A)	4.434	4.510	4.849
Потужність (W)	209.20	205.30	194.83

Результати розробки

Для адміністраторів сайту також доступний такий функціонал:

- Додавання, зміна та видалення електронних компонентів у каталозі
- Додавання, зміна та видалення категорій
- Перегляд та обробка скарг від користувачів

Скарги

Список скарг від користувачів щодо неповних чи некоректних даних компонентів

Компонент	Скарга	Дата відправлення	Статус	Дія
MC216	Потужність вказана неправильно	May 28, 2025, 1:43 a.m.	В обробці	Позначити як оброблено
TL490	Неправильно вказана категорія	May 28, 2025, 1:41 a.m.	В обробці	Позначити як оброблено
TL490	Зображення не відповідає дійсності	May 28, 2025, 1:39 a.m.	В обробці	Позначити як оброблено
AD849	Недостатньо детальний опис	May 28, 2025, 1:39 a.m.	В обробці	Позначити як оброблено
CRCWD60310K0FKFA	Не вказано тип корпусу	May 28, 2025, 1:37 a.m.	Оброблено	Позначити як в обробці

Результати розробки

Головна | Про проект | Категорії | Новини | Улюблені | Скарги

Привіт, component_admin!

Вийти

Список компонентів

Додай

BGA

V ≥

V ≤

A ≥

A ≤

W ≥

W ≤

Фільтрувати

Очистити

Додати компонент

Назва	Виробник	Категорія	Опис	Тип корпусу	Напруга	Струм	Потужність	Дія
AD849	Analog Devices	Діоди	Забезпечує функціональність в різноманітних електронних застосуваннях.	BGA	25.57	0.920	23.52	РедагуватиВидалити УлюбленіПорівняти
MC216	Infineon	Діоди	Важливий компонент у електричних схемах.	BGA	14.27	3.695	52.73	РедагуватиВидалити УлюбленіПорівняти
LM498	Maxim Integrated	Діоди	Електронний компонент у категорії "Діоди"	BGA	30.93	2.444	75.59	РедагуватиВидалити УлюбленіПорівняти

Результати розробки

Редагувати категорію

Назва категорії

Батьківська категорія

Опис (необов'язково)

Конденсатор є пасивним електронним компонентом і широко застосовується в електронних схемах для блокування постійного струму, пропускання змінного струму.

[Зберегти](#)

Редагувати компонент

Назва

Виробник

Категорія

Компактний SMD резистор серії CRCW з опором 10 кОм, точністю $\pm 1\%$, стандартним розміром 0603, підходить для широкого спектру застосувань у цифровій і аналоговій електроніці.

Опис

Тип корпусу

Робоча напруга (V)

Робочий струм (A)

Потужність (W)

Зображення (URL)

[Зберегти](#)

[Скасувати](#)

Категорії

- [Пасивні компоненти \(102\)](#)
 - [Резистори \(52\)](#) [\[редагувати\]](#) [\[видалити\]](#)
 - [Конденсатори \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
- [Активні \(150\)](#)
 - [Транзистори \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
 - [Діоди \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
 - [Інтегральні схеми \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
- [Живлення \(200\)](#)
 - [Джерела живлення \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
 - [Акумулятори \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
 - [Батареї \(50\)](#) [\[редагувати\]](#) [\[видалити\]](#)
- [Дисплеї \(50\)](#)

[Додати нову категорію](#)

Загальний алгоритм роботи системи

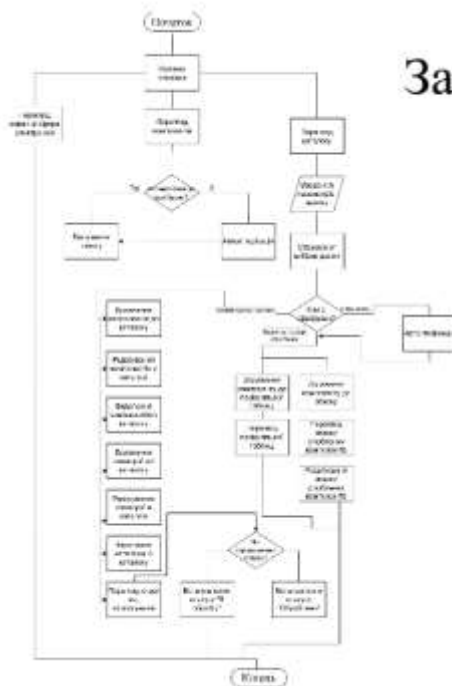
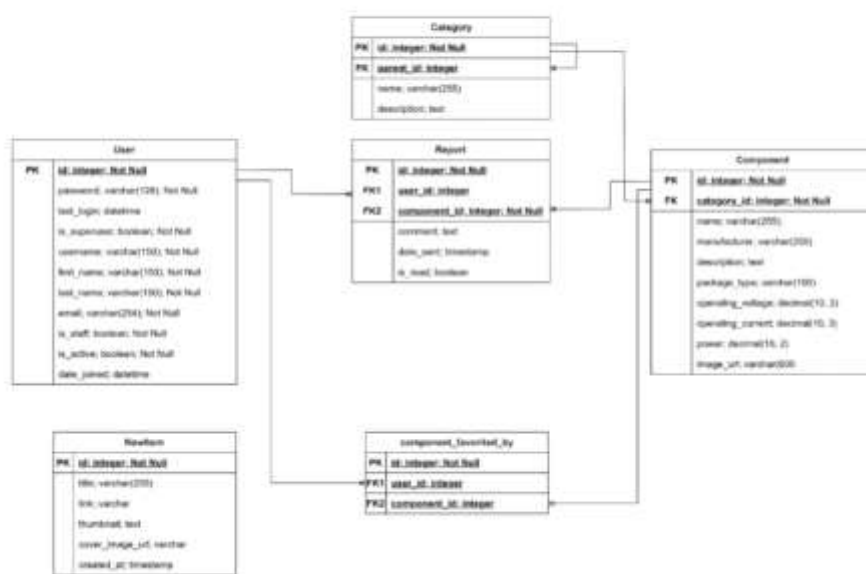


Схема бази даних



Алгоритм отримання даних відповідно до запиту

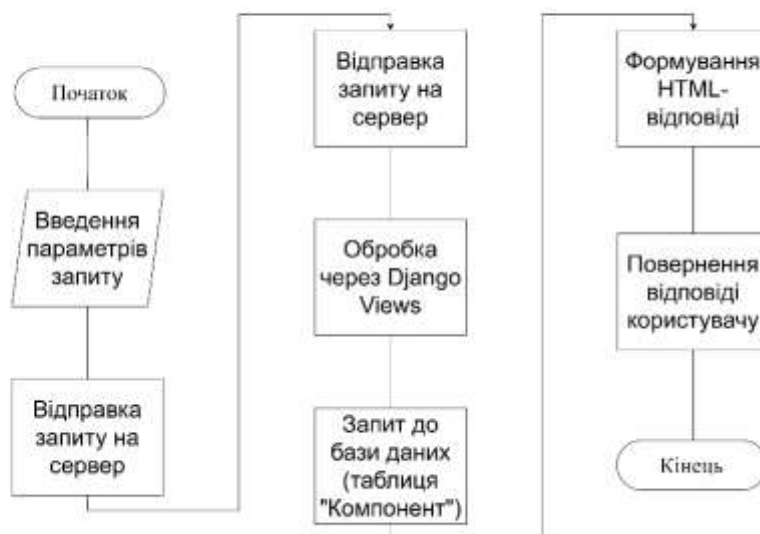


Схема а

Отже, у процесі виконання роботи було створено інформаційну систему для пошуку електронних компонентів, яка має низку переваг над популярними аналогами.

Серед інших переваг — відкритість, можливість легкої адаптації під специфічні потреби, а також відсутність комерційних обмежень чи нав'язаної реклами.

Система забезпечує ефективну взаємодію між користувачами та адміністраторами завдяки вбудованому механізму зворотного зв'язку. Завдяки цьому проєкт може бути особливо корисним у навчальних і дослідницьких цілях, а також для інженерів, виробників, студентів та інших технічних фахівців.