

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-60-02>

УДК 004.421 004.5

Овсяк Володимир Казимирович, д.т.н., професор

<https://orcid.org/0000-0001-9295-284X>

Турчак Володимир Романович, аспірант

<https://orcid.org/0009-0009-2907-1869>

Національний університет "Львівська політехніка", м. Львів, Україна

МОДЕЛЬ ІНТЕРФЕЙСУ КОРИСТУВАЧА СИСТЕМИ СОРТУВАННЯ СТРУКТУРОВАНИХ ДАНИХ

Овсяк В.К., Турчак В.Р. Модель інтерфейсу користувача системи сортування структурованих даних. Сортування табличних даних є ключовим етапом обробки структурованої інформації, що застосовується під час аналізу технічних списків, зокрема переліків охоронних сповіщувачів, або підготовки аналітичних матеріалів. У статті представлено методіку і модель розроблення інтерактивного вебінтерфейсу для сортування табличних структур, який забезпечує завантаження файлів (зокрема Excel) і впорядкування в браузері без потреби у встановленні додаткового програмного забезпечення. Інтерфейс підтримує режими ручного, покрокового (з проміжним відображенням результатів) і автоматичного сортування, даючи змогу задавати пріоритетні критерії: зростання, спадання, алфавітне впорядкування або фільтрацію за текстовим запитом. Інтерфейс складається з головного вікна, вікон вибору файлу, налаштування параметрів, пріоритетів, часткового збігу, вибору режимів упорядкування та фінального відображення результатів. Кожне вікно описано як впорядкована n -ка з елементами керування (кнопки, чекбокси, текстові поля). Модель деталізує логіку переходів між вікнами, механізми ініціалізації змінних і обробки помилок, що забезпечує інтуїтивну взаємодію з користувачем. Технічна реалізація виконана з використанням HTML, CSS, JavaScript і бібліотеки XLSX, а алгоритм сортування побудовано на принципах динамічно-наслідкового обмеженого переміщення даних. У практичній частині статті наведено фрагмент реалізації: демонструються скріншоти інтерфейсних вікон і приклад послідовної роботи системи на даних про охоронні сповіщувачі, що підтверджує ефективність моделі як для числових, так і для текстових полів. Тестування засвідчило гнучкість і продуктивність рішення, яке спрощує аналіз даних і є придатним для навчальних, офісних та прикладних застосувань. Наукова новизна полягає у формалізації взаємодії користувача з інтерфейсом сортування через множини та впорядковані кортежі, що описують стани системи й дії користувача, забезпечуючи повну логічну визначеність усіх переходів. Формальна модель забезпечує теоретичну основу для масштабування, а використання вебтехнологій гарантує доступність і простоту інтеграції. Порівняно з аналогами, розроблений інтерфейс вирізняється інтерактивністю, підтримкою текстових запитів і чіткою формалізацією логіки взаємодії, що підвищує його конкурентоспроможність.

Ключові слова: сортування даних, вебінтерфейс, алгоритм сортування, інтерактивність, аналіз даних, охоронні сповіщувачі, математична модель, частковий збіг.

Ovsjak V., Turchak V. User Interface Model for Sorting Structured Data. Sorting tabular data is a critical stage in processing structured information, applied in the analysis of technical lists, such as security sensor inventories, or the preparation of analytical materials. This article presents a methodology and model for developing an interactive web interface for sorting tabular structures, enabling file upload (e.g., Excel) and in-browser sorting without additional software. The interface supports manual, step-by-step (with intermediate result display), and automatic sorting modes, allowing prioritization by ascending, descending, alphabetical order, or text-based filtering. The interface consists of a main window, file selection, parameter settings, priority settings, partial match, sorting mode selection, and final result display windows. Each window is described as an ordered n -tuple with control elements (buttons, checkboxes, text fields). The model details window transition logic, variable initialization, and error handling, ensuring intuitive user interaction. The technical implementation uses HTML, CSS, JavaScript, and the XLSX library, with the sorting algorithm based on dynamic-constrained data movement principles. The practical section demonstrates interface window screenshots and the system's sequential operation on security sensor data, confirming its effectiveness for numeric and text fields. Testing revealed the solution's flexibility and performance, suitable for educational, office, and applied use. The scientific novelty lies in formalizing user interaction with the sorting interface through sets and ordered tuples, describing system states and user actions, ensuring complete logical determinacy of transitions. The formal model provides a theoretical basis for scalability, while web technologies ensure accessibility and ease of integration. Compared to analogs, the developed interface stands out for its interactivity, support for text queries, and clear formalization of interaction logic, enhancing its competitiveness.

Keywords: data sorting, web interface, sorting algorithm, interactivity, data analysis, security sensors, mathematical model, partial match.

Вступ

Опрацювання структурованих даних у форматі таблиць є невід'ємною складовою цифрового документообігу, технічного моніторингу та інформаційно-аналітичних систем. Із розширенням обсягів даних зростає потреба в засобах їх ефективного впорядкування, особливо у вебсередовищі, де ключовими чинниками є доступність, інтерактивність і зручна взаємодія з користувачем. Особливу актуальність мають інтерфейси, що не потребують встановлення

програмного забезпечення, підтримують кросплатформеність і функціонують безпосередньо в браузері.

У контексті сучасних вимог до UX/UI (User Experience / User Interface) та HCI (Human–Computer Interaction) важливо забезпечити не лише високу швидкість алгоритмів, а й прозору логіку взаємодії, адаптивність інтерфейсу та можливість поетапного контролю дій користувача. У прикладних задачах – від впорядкування технічних переліків (наприклад, охоронних сповіщувачів) до візуалізації великих інформаційних масивів – визначальними є функції багатокритеріального сортування, текстової фільтрації та проміжного відображення результатів. Водночас сучасні бібліотеки, як-от SheetJS, PapaParse, Handsontable, хоча й забезпечують базову обробку табличних структур, не завжди придатні для реалізації покрокової логіки або потребують складного ручного налаштування.

Дослідження орієнтоване на побудову інтерфейсу, що дозволяє користувачеві задавати пріоритети, обирати режим сортування (ручний, покроковий, автоматичний) і застосовувати фільтрацію за текстовим запитом. Основою створеного рішення є математичні моделі на базі теорії множин.

Метою дослідження є розроблення, реалізація та аналіз інтерактивного вебінтерфейсу для сортування табличних даних, що підтримує ручний, покроковий і автоматичний режими впорядкування з урахуванням числових, рядкових і датованих значень, а також варіантів порівняння – зростання, спадання або частковий збіг.

Об'єктом дослідження є процеси взаємодії користувача з системою сортування структурованих даних.

Предметом дослідження є математичні моделі структур даних та інтерфейсу користувача системи сортування структурованих даних.

Для досягнення поставленої мети потрібно розв'язати такі завдання:

1. Проаналізувати сучасні методи сортування табличних даних у вебінтерфейсах, виявити обмеження бібліотек SheetJS, PapaParse, Handsontable щодо покрокового відображення і гнучкості обробки.

2. Побудувати математичні моделі структури даних та інтерфейсу і реалізувати вебзастосунок на базі HTML, CSS, JavaScript і бібліотеки XLSX, протестувавши його на тестовому наборі даних охоронних сповіщувачів.

3. Провести порівняльний аналіз ефективності розроблених рішень за критеріями швидкодії, зручності використання та гнучкості, підкресливши переваги формалізованої структури й покрокового режиму сортування.

Аналіз останніх досліджень і публікацій.

У сфері інформаційних технологій, що поєднує алгоритмічну оптимізацію, обробку структурованих даних і розробку інтерактивних інтерфейсів, сортування табличних даних у вебінтерфейсах є важливим напрямом досліджень. Сучасні роботи зосереджені на інтеграції табличних форматів (Excel, CSV, JSON) у браузерні середовища, забезпеченні кросплатформності та підтримці функцій фільтрації й візуалізації [1, 2].

Значна частина досліджень присвячена JavaScript-бібліотекам для роботи з табличними даними. Бібліотека SheetJS (XLSX) ефективно обробляє великі Excel-файли, підтримуючи базове сортування за числовими й текстовими критеріями. Її перевагою є швидкий імпорт і експорт даних, але вона не забезпечує покрокового відображення чи гнучкої текстової фільтрації. PapaParse, орієнтована на CSV, вирізняється швидким парсингом і сортуванням, але обмежена у візуалізації проміжних результатів і налаштуванні багатокритеріальних пріоритетів. Handsontable імітує функціонал таблиць, дозволяючи сортувати за кількома стовпцями завдяки зручному API, але її можливості текстової фільтрації та інтерактивного покрокового сортування недостатні для аналітичних завдань [6, 8].

Інший напрям досліджень стосується алгоритмів сортування, адаптованих для вебзастосунків. Класичні алгоритми, такі як QuickSort і MergeSort, оптимізовані для швидкості, є ефективними для невізуальних задач, але не підтримують інтерактивне відображення чи збереження часткової структури даних. Гібридні алгоритми, що поєднують сортування з фільтрацією, пропонують часткове вирішення, але рідко інтегруються з вебінтерфейсами, які дозволяють динамічне налаштування критеріїв [3, 5].

Дослідження групового сортування, яке враховує колективні пріоритети, пропонують нові підходи до багатокритеріальних задач, однак такі методи не адаптовані до браузерного середовища з текстовою фільтрацією [4, 7]. Також важливими є алгоритми, що поєднують багатокритеріальність із динамічним налаштуванням параметрів сортування.

Ці алгоритмічні підходи тісно пов'язані з формалізацією інтерфейсів для опрацювання даних. Моделі на основі теорії множин і впорядкованих кортежів описують структури даних, але їх застосування до вебінтерфейсів обмежене. Бракує формальних моделей, які інтегрують логіку вікон, обробку помилок і підтримку режимів сортування (ручний, покроковий, автоматичний) [8].

Дослідження в сфері HCI, а також UX/UI, доводять важливість адаптивності інтерфейсів і автоматизованого врахування дій користувача [6, 9]. Актуальні праці аналізують ефективність адаптацій у табличних структурах і вплив інтерфейсного дизайну на продуктивність.

Аналіз літератури свідчить про прогалину в розробці вебінтерфейсів, які поєднують багатокритеріальне сортування, покрокове відображення, текстову фільтрацію та формалізацію логіки. Обмеження бібліотек SheetJS, PapaParse і Handsontable полягають у слабкій інтерактивності та відсутності підтримки складних запитів. Алгоритми сортування не враховують потребу в динамічних пріоритетах і частковому збереженні структури. На відміну від аналогів, створене рішення забезпечує три режими сортування, чітку формалізацію вікон і придатність для аналізу даних, таких як переліки охоронних сповіщувачів, у навчальних, офісних і технічних сценаріях [10].

Методи дослідження

Формалізація вхідних даних, та створення математичної моделі інтерфейсу користувача виконано застосуванням упорядкованих n – нок теорії множин;

Програмна реалізація виконана із використанням HTML5, CSS, JavaScript та бібліотеки XLSX (SheetJS), що забезпечило обробку Excel-файлів у браузері без додаткового програмного забезпечення. Тестування проводилося на даних про охоронні сповіщувачі, що включають числові та текстові поля. Оцінено коректність переходів, стабільність сортування, час виконання та використання пам'яті у різних браузерах (Chrome, Firefox, Edge), що підтвердило продуктивність створених моделей.

1. Формалізація вхідних даних, та створення математичної моделі інтерфейсу користувача

1.1. Модель структурованих даних

Під час інтерактивного сортування користувач працює з базою даних, яка в даному випадку представлена таблицею структурованих даних – наприклад, переліком охоронних сповіщувачів. Дані можуть мати довільну кількість об'єктів (рядків) і характеристик (стовпців), тому їх подаємо у вигляді прямокутної матриці D , де кожен рядок відповідає окремому об'єкту, а кожен стовпець – фіксованому атрибуту.

Модель структури даних має такий вигляд:

$$D = \begin{pmatrix} x_{0,0} & x_{0,1} & x_{0,2} & \dots & x_{0,m-1} \\ x_{1,0} & x_{1,1} & x_{1,2} & \dots & x_{1,m-1} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n-1,0} & x_{n-1,1} & x_{n-1,2} & \dots & x_{n-1,m-1} \end{pmatrix} \quad (1)$$

де n – кількість рядків (об'єктів) у таблиці,

m – кількість атрибутів (стовпців),

$x_{j,i}$ значення i – го атрибута для j – го об'єкта, $j \in \{0, \dots, n-1\}$, $i \in \{0, \dots, m-1\}$.

Модель структурованих даних засобами теорії множин описана такою впорядкованою n -кою:

$$D = (r_0, r_1, r_2, \dots, r_{n-1}) \quad (2)$$

де кожен запис r_j (тобто рядок таблиці) представляє окремий об'єкт у вигляді впорядкованої m -ки значень j -го об'єкта:

$$r_j = (x_{j,0}, x_{j,1}, x_{j,2}, \dots, x_{j,m-1}) \quad (3)$$

Модель є основою для побудови процедур сортування за кількома критеріями.

1.2. Модель головного вікна

Головне вікно є впорядкованою четвіркою:

$$(c_0, t, z, d) \quad (4)$$

де c_0 – перший, t – другий, z – третій і d – четвертий елемент впорядкованої четвірки. Записавши кожен елемент впорядкованої четвірки окремим рядком, отримаємо:

$$\begin{pmatrix} c_0, \\ t, \\ z, \\ d \end{pmatrix} \quad (5)$$

Нехай елементи t , z та d є такими упорядкованими n -ками (vf, c_1) , (r_j, k) та $(e, e+1, a)$, відповідно. Виконавши підстановки у впорядковану четвірку (5) значення t , z та d , отримуємо такий вираз (6):

$$\begin{pmatrix} c_0, \\ (vf, c_1), \\ (r_j, k), \\ (e, e+1, a) \end{pmatrix} \quad (6)$$

Нехай c_0 – стала, якою є слово "Сортування";

vf – змінна/кнопка вибору файлу, натискання якої відкриває вікно вибору файлу;

c_1 – повідомлення про помилку стала ("Файл не вибрано") яке з'являється, якщо змінна vf не ініціалізована;

$r_j = m_j$ – графічний об'єкт у списку, який відповідає об'єкту $r_j \in D$;

k – змінна критеріїв сортування;

e – режим сортування;

$e+1$ – режим "Наступний крок";

a – режим автоматичного сортування.

Вираз (6) є моделлю головного вікна системи.

1.3. Модель вікна вибору файлу

Модель вікна вибору файлу описується впорядкованою шестіркою:

$$(f_0, f_1, f_2, f_3, f_4, f_5) \quad (7)$$

де f_0 – перший, f_1 – другий, f_2 – третій, f_3 – четвертий, f_4 – п'ятий, f_5 – шостий елемент впорядкованої шестірки.

Записавши кожен елемент впорядкованої шестірки окремим рядком, отримаємо:

$$\begin{pmatrix} f_0, \\ f_1, \\ f_2, \\ f_3, \\ f_4, \\ f_5 \end{pmatrix} \quad (8)$$

Нехай елементи f_1, f_2, f_3, f_4, f_5 є такими впорядкованими n -ками:

$$\begin{aligned} f_1 &= (p_1, g, p_2), \\ f_2 &= (p_3, p_4), \\ f_3 &= (dir, o, pp), \\ f_4 &= (p_5, h), \\ f_5 &= (b_1, b_2, c_3) \end{aligned}$$

Виконавши підстановки у впорядковану шестірку (8), отримаємо (9):

$$\left(\begin{array}{c} f_0, \\ (p_1, g, p_2) \\ (p_3, p_4), \\ (dir, o, pp) \\ (p_5, h), \\ (b_1, b_2, c_3) \end{array} \right) \quad (9)$$

де f_0 – стала, що відображає “Відкриття файлу”;
 p_1 – панель навігації у вигляді набору стрілок ("назад", "вперед", "нешодавнє розташування", "уверх") для навігації по директоріях;
 g – поточний шлях до файлу ("Цей ПК > Desktop > BAZA");
 p_2 – поле пошуку файлу;
 p_3 – панель навігації з базовими діями ("Упорядкувати", "Створити");
 p_4 – панель навігації ("додаткові можливості", "відобразити область перегляду", "отримати довідку");
 dir – змінна поточної директорії;
 o – список об'єктів у директорії (файлів і папок);
 pp – попередній перегляд вмісту файлу;
 p_5 – поле виведення назви файлу;
 h – тип файлів, вибраний у випадуючому списку (наприклад: "Excel-файли");
 b_1 – кнопка "Відкрити";
 b_2 – кнопка "Скасувати";
 c_3 – якщо вибраний файл не є таблицею або має неправильний формат, виводиться повідомлення "Некоректний формат файлу".

Вираз (9) є моделлю вікна вибору файлу.

1.4. Модель вікна з таблицею та панеллю параметрів

Модель вікна описана такою впорядкованою четвіркою:

$$(p_0, p_1, p_2, p_3) \quad (10)$$

де p_0 – перший, p_1 – другий, p_2 – третій, p_3 – четвертий елемент впорядкованої четвірки. Записавши кожен елемент впорядкованої четвірки окремим рядком, отримаємо:

$$\left(\begin{array}{c} p_0, \\ p_1, \\ p_2, \\ p_3 \end{array} \right) \quad (11)$$

Нехай елементи p_1, p_2, p_3 мають внутрішню структуру у вигляді впорядкованих n -нок:
 $p_0 = c_0, p_1 = vf, p_2 = (L, R, c_4), p_3 = (e, e+1, a)$.
 Підставивши значення у (11), отримаємо (12):

$$\begin{pmatrix} c_0, \\ vf, \\ (L, R, c_4) \\ (e, e + 1, a) \end{pmatrix} \quad (12)$$

де c_0 – стала, якою є слово "Сортування";

vf – змінна вибору файлу;

$L \subset D$ – список виведених об'єктів рядків матриці D ;

$R = \{ R_0, \dots, R_{m-1} \}$ – множина критеріїв кожен з яких $R_i = (\delta_i, \rho_i, pr_i, \tau_i)$, $\delta_i \in \{0, 1\}$, ρ_i тип сортування {"макс", "мін", "A-Z", "Z-A"}, $pr_i \in N$ – пріоритет, $\tau_i \in \{0, 1\}$ – активація часткового збігу;

c_4 – Якщо $\forall_i: \delta_i = 0$, виводиться повідомлення: "Виберіть хоча б один критерій",

e – режим сортування;

$e + 1$ – режим "Наступний крок";

a – режим автоматичного сортування.

1.5. Модель вікна з панеллю та пріоритетами

На першому етапі модель панелі пріоритетів описується впорядкованою четвіркою:

$$(g_0, g_1, g_2, g_3) \quad (13)$$

де кожен елемент $g_0 - g_3$ описує окремий логічний компонент інтерфейсу. Записавши їх у вертикальній формі, отримаємо:

$$\begin{pmatrix} g_0, \\ g_1, \\ g_2, \\ g_3 \end{pmatrix} \quad (14)$$

де $g_0 - c_0$ заголовок вікна, текстове поле зі сталою «Сортування»;

$g_1 - vf$ змінна вибору файлу;

$g_2 - (L, R)$ множина об'єктів (рядків) з таблиці D , що відображається у списку сповіщувачів зліва, а R множина активних критеріїв, де кожен критерій R_i подається як впорядкована n -нка:

$$R_i = (\delta_i, \rho_i, pr_i, \tau_i), \quad (15)$$

де $\delta_i \in \{0, 1\}$ – стан чекбокса (1 – активний, 0 – неактивний),

ρ_i тип сортування {"макс", "мін", "A-Z", "Z-A"},

$pr_i \in N$ – порядковий пріоритет критерію у процесі сортування;

$\tau_i \in \{0, 1\}$ – активація часткового збігу,

$g_3 - (e, e + 1, a)$ трійка логічних змінних, що відповідають режимам взаємодії:

e – ручне сортування,

$e + 1$ – режим покрокового сортування,

a – автоматичний запуск сортування.

Модель вікна з панеллю та пріоритетами має вигляд (16):

$$\begin{pmatrix} c_0, \\ vf, \\ (L, R), \\ (e, e + 1, a) \end{pmatrix} \quad (16)$$

Після деактивації критерію ($\delta_i = 0$) система автоматично переобчислює значення пріоритетів pr_j для решти активних критеріїв, зберігаючи їхній відносний порядок. Якщо активовано текстовий критерій з підтримкою часткового збігу ($\tau_i = 1$), система відкриває модальне вікно для налаштування умов фільтрації.

1.6. Модель вікна часткового збігу

Вікно часткового збігу активується при виборі опції «порівняння з запитом» у випадіючому списку режиму сортування, що з'являється після активації чекбокса (наприклад, «Сповіщувач руху охоронний») у правій панелі для критерію R_i з $\tau_i = 1$. Це дає змогу користувачеві ввести текст для фільтрації/порівняння.

Модель цього модального вікна описується впорядкованою четвіркою:

$$(u_0, u_1, u_2, u_3) \tag{17}$$

де u_0 – перший, u_1 – другий, u_2 – третій, u_3 – четвертий елемент четвірки. Записавши кожен елемент окремим рядком, отримаємо:

$$\begin{pmatrix} u_0, \\ u_1, \\ u_2, \\ u_3 \end{pmatrix} \tag{18}$$

нехай u_0 – заголовок вікна ("Введіть значення для порівняння");

$u_1 = \Sigma$ – поле для введення рядка (змінна введення, де Σ – довільний текст);

$u_2 \in V = \{x_{\{j,i\}} \mid j = 0, \dots, n-1\}$, де i – стовпець, вибраний у R_i з $\tau_i = 1$;

$u_3 = (b_1, b_2)$ – кнопки керування:

b_1 – "ОК" – підтвердження введення,

b_2 – "×" – закриття вікна без дії.

c_2 – Якщо жоден елемент із V не містить підрядка Σ , з'являється повідомлення "Немає збігів".

Значення, введене в поле Σ , зіставляється з множиною V за умовами часткового або повного збігу залежно від режиму сортування. У разі підтвердження користувачем натисканням кнопки b_1 , введене значення передається до механізму сортування як запит.

Модель вікна часткового збігу виглядає так (19):

$$\begin{pmatrix} m_0, \\ (i), \\ (V), \\ (b_1, b_2, c_2) \end{pmatrix} \tag{19}$$

1.7. Модель вікна панелі режимів упорядкування

Панель режимів упорядкування реалізує лише логіку вибору режиму взаємодії користувача, на відміну від панелі пріоритетів, де реалізовано логіку вибору і впорядкування критеріїв.

Панель режимів упорядкування описується впорядкованою четвіркою:

$$(s_0, s_1, s_2, s_3) \tag{20}$$

де s_0 – перший, s_1 – другий, s_2 – третій, s_3 – четвертий елемент четвірки. Записавши кожен елемент окремим рядком, отримаємо:

$$\begin{pmatrix} s, \\ s_1, \\ s_2, \\ s_3 \end{pmatrix} \tag{21}$$

де кожен елемент представляє окрему функціональну область вікна:

s_0 – заголовок вікна, фіксується сталою «Сортування»;

s_1 – vf змінна вибору файлу;

$s_2 = (L, R)$ множина об'єктів (рядків) з таблиці D , що відображається у списку сповіщувачів зліва, а R множина активних критеріїв, де кожен критерій R_i подається як впорядкована n -енка;

$s_3 = (e, e+1, a)$ логічні змінні, що відповідають режимам взаємодії:

e – ручне сортування,

$e+1$ – режим покрокового сортування,

a – автоматичний запуск сортування.

Підставивши значення у (21), отримаємо остаточною модель (22):

$$\begin{pmatrix} s_0, \\ vf, \\ (L, R), \\ (e, e + 1, a) \end{pmatrix} \quad (22)$$

1.8. Модель вікна фінального результату

Модель вікна описується впорядкованою трійкою:

$$(t_0, t_1, t_2) \quad (23)$$

де t_0, t_1, t_2 – компоненти, що визначають зміст і структуру фінального вікна результатів.

Записавши елементи по рядках, отримаємо:

$$\begin{pmatrix} t_0, \\ t_1, \\ t_2 \end{pmatrix} \quad (24)$$

Уточнюємо зміст кожного компонента так:

$t_0 = w_0$ – заголовок вікна, сталий напис "Фінальні результати сортування";

$t_1 = H$ – заголовки стовпців таблиці (відповідають набору критеріїв, виділені кольором пріоритетні атрибути).

Формально $H = \{h_0, h_1, \dots, h_{m-1}\}$, де кожен h_i відповідає окремому атрибуту;

$t_2 = D'$ – впорядкована множина рядків, результат застосування заданих критеріїв і пріоритетів до початкових даних.

Кожен рядок – це впорядкована множина значень атрибутів:

$$d_j = (x_{j,0}, x_{j,1}, \dots, x_{j,m-1}),$$

де $j = 0, \dots, n-1$.

Остаточна модель вікна має такий вигляд (25):

$$\begin{pmatrix} w_0, \\ H, \\ D' \end{pmatrix} \quad (25)$$

2. Верифікація моделі інтерфейсу

Розроблений вебінтерфейс для інтерактивного сортування табличних даних забезпечує інтуїтивну взаємодію та кросплатформенну підтримку. Реалізований за допомогою HTML5, CSS3, JavaScript і бібліотеки SheetJS, він працює на пристроях із сучасними браузерами (Google Chrome, Mozilla Firefox, Safari, Edge тощо) без потреби в додатковому програмному забезпеченні. Він також пропонує зручний процес упорядкування даних, що демонструється на прикладі таблиці з інформацією про охоронні сповіщувачі. Кожен етап виконується плавно, забезпечуючи інтуїтивну взаємодію користувача з системою. Нижче наведено детальний опис послідовності дій для сортування з використанням різних режимів.

Процес розпочинається з відкриття вебсторінки в браузері Google Chrome, де користувач бачить головне вікно з назвою «Сортування» та панеллю кнопок (рис. 1).

Сортування

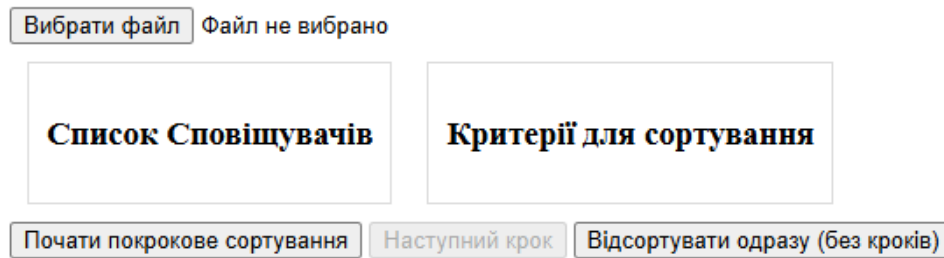


Рис. 1. Головне вікно

Натискання кнопки "Вибрати файл" відкриває вікно вибору файлу, яке виглядає як стандартне системне вікно, знайоме кожному користувачеві. Воно містить панель навігації зі стрілками ("назад", "вперед", "нещодавні розташування") та відображенням поточного шляху, наприклад, "Цей ПК > Робочий стіл > BAZA". Поле пошуку дозволяє швидко знайти потрібний файл, а список файлів і папок відображає вміст поточної директорії. Функція попереднього перегляду демонструє структуру обраного Excel-файлу, такого як "охорону_sprov.xlsx". У випадаючому списку користувач може обрати тип файлів ("Excel-файли"), щоб звузити пошук. Кнопки "Відкрити" та "Скасувати" завершують цей етап: "Відкрити" зчитує дані за допомогою бібліотеки SheetJS і переводить до наступного вікна, а "Скасувати" повертає до головного екрана. Якщо обрано файл невідповідного формату, наприклад, PDF, з'являється повідомлення: "Файл не вибрано". (рис. 2).

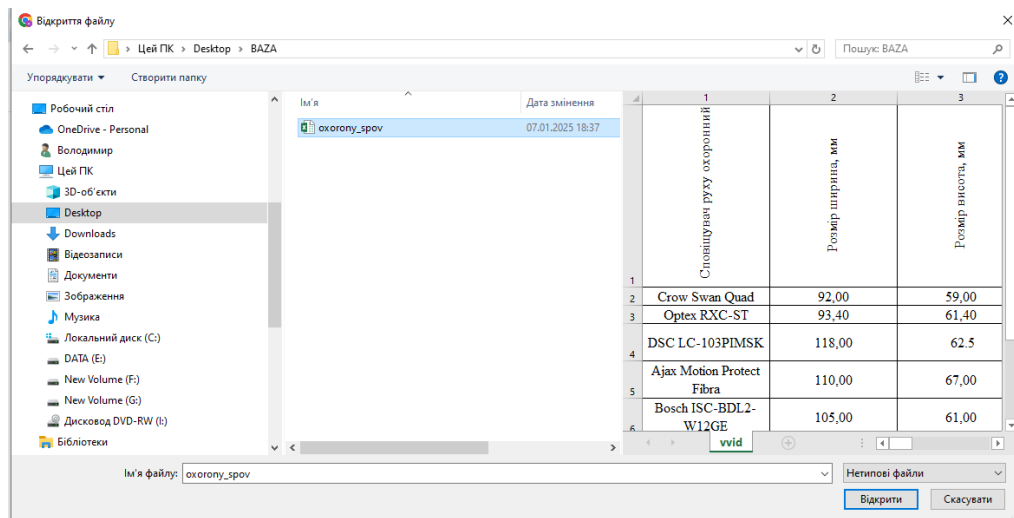


Рис. 2. Вікно вибору файлу

Після завантаження файлу відкривається вікно, де структуровані дані стають доступними для перегляду та обробки. Ліворуч відображається таблиця з рядками файлу, наприклад перелік охоронних сповіщувачів. Праворуч розташована панель для вибору критеріїв сортування, а вгорі – заголовок "Сортування", який підтверджує поточний етап роботи. Якщо потрібно замінити файл, кнопка "Вибрати файл" дозволяє повернутися до попереднього етапу (рис. 3).

Сортування

Вибрати файл oхороны_sprov.xlsx

Список Сповіщувачів

Crow Swan Quad
Optex RXC-ST
DSC LC-103PIMSK
Ajax Motion Protect Fibra
Bosch ISC-BDL2-W12GE
Satel SLIM-PIR
Pyronix COLT10DL
GSN Patrol-903
Pyronix COLT QPI
Satel GRAPHITE

Критерії для сортування

Сповіщувач руху охоронний За абеткою (A-Z) ▼

Розмір ширина, мм Максимальне значення ▼

Розмір висота, мм Максимальне значення ▼

Розмір глибина, мм Максимальне значення ▼

Ціна Максимальне значення ▼

Робоча напруга мінімальна В Максимальне значення ▼

Робоча напруга максимальна В Максимальне значення ▼

Споживання: режим очікування, мА Максимальне значення ▼

Споживання: режим активний, мА Максимальне значення ▼

Оптимальна висота установки мінімальна Максимальне значення ▼

Оптимальна висота установки максимальна Максимальне значення ▼

Дальність виявлення руху Максимальне значення ▼

Кут огляду, С° Максимальне значення ▼

Робоча температура мінімальна С° Максимальне значення ▼

Робоча температура максимальна С° Максимальне значення ▼

Імунітет до тварин (вага) кг Максимальне значення ▼

Спосіб виявлення руху: інфрачервоний, мікрохвильовий За абеткою (A-Z) ▼

Почати покрокове сортування Наступний крок Відсортувати одразу (без кроків)

Рис. 3. Вікно з таблицею та панеллю параметрів

У панелі критеріїв користувач активує потрібні параметри, ставлячи галочки біля потрібних пунктів і обирає тип сортування: зростання (від меншого до більшого), спадання чи алфавітний порядок (A–Z або Z–A). Якщо користувач намагається продовжити без вибору критеріїв, система видає повідомлення: "Оберіть хоча б один критерій для сортування". Це вікно підтримує всі режими сортування, забезпечуючи гнучкість для роботи з числовими та текстовими даними. (рис. 4).

Сортування

Вибрати файл oхороны_sprov.xlsx

Список Сповіщувачів

Crow Swan Quad
Optex RXC-ST
DSC LC-103PIMSK
Ajax Motion Protect Fibra
Bosch ISC-BDL2-W12GE
Satel SLIM-PIR
Pyronix COLT10DL
GSN Patrol-903
Pyronix COLT QPI
Satel GRAPHITE

Критерії для сортування

Сповіщувач руху охоронний За абеткою (A-Z) ▼

Розмір ширина, мм Максимальне значення ▼

Розмір висота, мм Максимальне значення ▼

Розмір глибина, мм Максимальне значення ▼

Ціна Максимальне значення ▼ **Пріоритет: 1**

Робоча напруга мінімальна В Максимальне значення ▼

Робоча температура мінімальна В Максимальне значення ▼

Споживання: режим очікування, мА Максимальне значення ▼

Споживання: режим активний, мА Максимальне значення ▼

Оптимальна висота установки мінімальна Максимальне значення ▼

Оптимальна висота установки максимальна Максимальне значення ▼

Дальність виявлення руху Максимальне значення ▼

Кут огляду, С° Максимальне значення ▼

Робоча температура мінімальна С° Максимальне значення ▼

Робоча температура максимальна С° Максимальне значення ▼

Імунітет до тварин (вага) кг Максимальне значення ▼

Спосіб виявлення руху: інфрачервоний, мікрохвильовий За абеткою (A-Z) ▼

Почати покрокове сортування Наступний крок Відсортувати одразу (без кроків)

Рис. 4. Вікно з таблицею та пріоритетами

Користувач активує прапорець для стовпця «Ціна» і обирає напрямок сортування «Мінімальне значення» у випадіючому списку, після чого система присвоює цьому критерію пріоритет 1, відображаючи його синім текстом.

Для текстових критеріїв по типу "Сповіщувач руху охоронний" доступна опція "Порівняння з запитом", яка відкриває спеціальне модальне вікно для введення тексту (рис. 5).

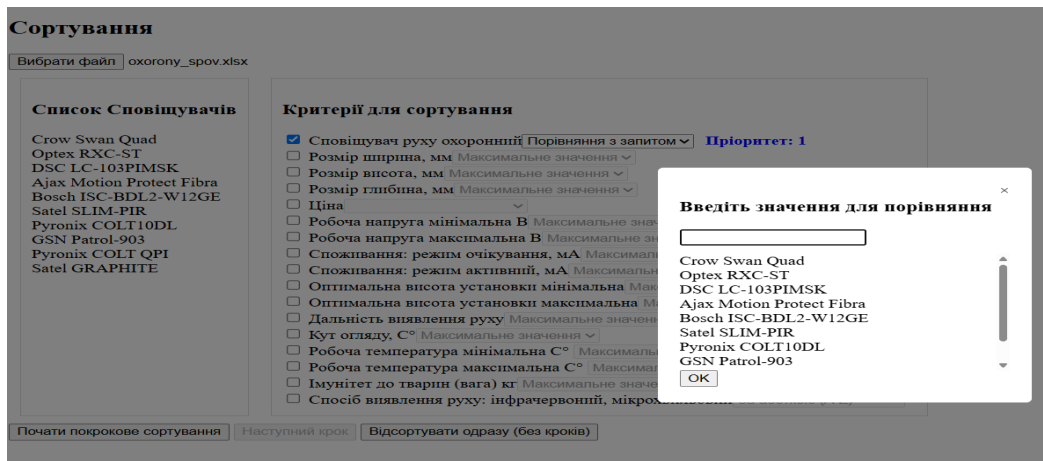


Рис. 5. Модальне вікно часткового збігу

У цьому вікні з напівпрозорим тлом і білим блоком користувач вводить запит «.....» у текстове поле або обирає варіант «Сповіщувач руху» зі списку внизу, і підтверджує натисканням «ОК». Вікно закривається, а критерій оновлюється в панелі пріоритетів, завершуючи налаштування (рис. 6).

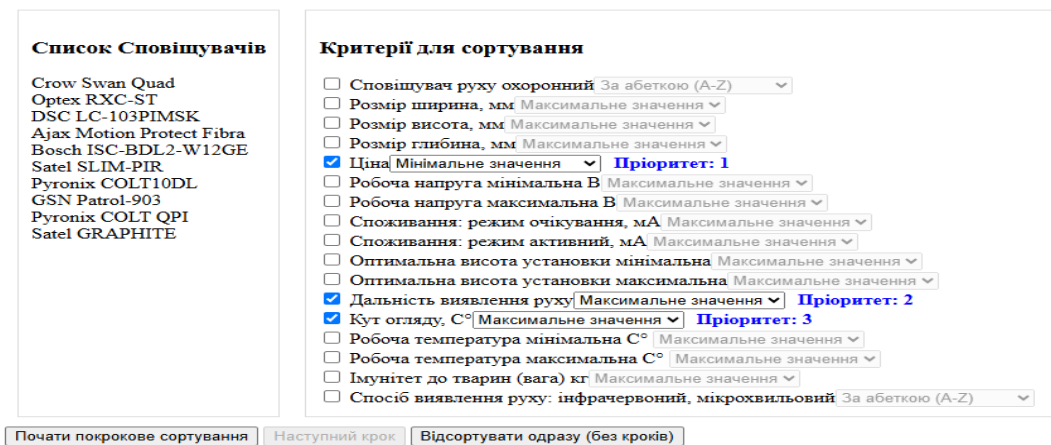


Рис. 6. Вікно панелі режимів упорядкування

якож користувач може додати ще критерії, наприклад другий критерій – «Дальність виявлення руху» з напрямком сортування «Максимальне значення» (пріоритет 2), та третій критерій – «Кут огляду» (пріоритет 3), з напрямком сортування «Максимальне значення» що готує інтерфейс до подальшої роботи. При деактивації критерію система автоматично перераховує пріоритети, зберігаючи їх логічний порядок.

Наступним кроком користувач обирає покроковий режим, натискаючи «Почати покрокове сортування». Оскільки критерії визначені, система застосовує перший критерій («ціна», «мінімальне значення»), сортує таблицю та підсвічуючи стовпець «Ціна» жовтим. Користувач натискає «Наступний крок», і другий критерій («Дальність виявлення руху», «Максимальне значення») оновлює таблицю з новим підсвічуванням. На третьому кроці («Кут огляду», «Максимальне значення»), після чого кнопка «Наступний крок» деактивується. Також користувач зможе одразу отримати кінцевий результат – обравши «Відсортувати одразу(без кроків)».

На завершальному етапі відображається вікно фінальної таблиці (рис. 7)

Сортування

Вибрати файл | [список_проектів](#)

Список Співшувачів

- Crow Swan Quad
- Ortek RXC-ST
- DSC LC-103PMSK
- Ajax Motion Protect Fibra
- Bosch ISC-BDL2-W1DGE
- Sanei SLIM-FIR
- Рytonix COLTI0DL
- GSN Patrol-903
- Рytonix COLT QPI
- Sanei GRAPHITE

Критерії для сортування

- Співшувач руху: **окорочений** За алфавітом (A-Z)
- Розмір ширини, мм Максимальне значення
- Розмір висоти, мм Максимальне значення
- Розмір глибини, мм Максимальне значення
- Ціна** Мінімальне значення **Пріоритет: 1**
- Робоча напруга мінімальна В Максимальне значення
- Робоча напруга максимальна В Максимальне значення
- Споживання: режим очікування, мА Максимальне значення
- Споживання: режим активний, мА Максимальне значення
- Оптимальна висота установки мінімальна Максимальне значення
- Оптимальна висота установки максимальна Максимальне значення
- Дальність виявлення руху** Максимальне значення **Пріоритет: 2**
- Кут огляду, С°** Максимальне значення **Пріоритет: 3**
- Робоча температура мінімальна С° Максимальне значення
- Робоча температура максимальна С° Максимальне значення
- Вугнет до тварин (bkg) кг Максимальне значення
- Способі виявлення руху: **інфрачервоний, мультиспектральний** За алфавітом (A-Z)

[Почати покрокове сортування](#) | [Наступний крок](#) | [Відсортувати одразу \(без кроків\)](#)

Фінальний результат сортування

Співшувач руху окорочений	Розмір ширини, мм	Розмір висоти, мм	Розмір глибини, мм	Ціна (P1)	Робоча напруга мінімальна В	Робоча напруга максимальна В	Споживання: режим очікування, мА	Споживання: режим активний, мА	Оптимальна висота установки мінімальна	Оптимальна висота установки максимальна	Дальність виявлення руху (P2)	Кут огляду, С° (P3)
GSN Patrol-903	54	33	99	714	8.5	16	9.7	10.5	2	2.5	15.6	125
Crow Swan Quad	92	59	37	663	8.2	16	8	10	1.8	2.4	18	110
Sanei GRAPHITE	63	96	49	980	10.2	13.8	10	13	1.8	2.4	20	90
Sanei SLIM-FIR	62	137	42	771	10.2	13.8	8	23	2.4	4	20	90
DSC LC-103PMSK	118	62.5	41	950	9.6	16	20	25	1.8	2.4	15	90
Рytonix COLT QPI	75	52	44	431	9	16	15	15	1.8	2.4	15	90
Bosch ISC-BDL2-W1DGE	105	61	44	1715	9	15	8	10	2.2	2.75	12	90
Рytonix COLTI0DL	82	50	41	369	9	16	15	15	2	2.4	10	90
Ajax Motion Protect Fibra	110	67	62	1124	24	24	0.05	0.05	1.8	2.4	12	88.5
Ortek RXC-ST	93.4	61.4	46	816	9.5	16	8	11	1.5	2.4	12	85

Рис. 7. Вікно фінального результату

під заголовком «Фінальні результати сортування». Таблиця показує співшувачі, відсортовані за вибраними критеріями. Стовпці-критерії підсвічено жовтим, у заголовках вказано пріоритети. Після виконання сортування (незалежно від обраного режиму – покрокового або миттєвого) користувач бачить на екрані фінальну таблицю з результатом. У цей момент процес сортування формально завершується, але система дозволяє здійснити повторні дії або змінити параметри без перезавантаження сторінки.

Кнопки «Обрати файл», «Почати покрокове сортування» і «Відсортувати одразу» залишаються активними, даючи змогу завантажити новий файл, змінити критерії або повторити сортування. При виборі нового файлу налаштування скидаються, а інтерфейс оновлюється. Завершення сесії (закриття сторінки) призводить до втрати даних, оскільки обробка здійснюється в оперативній пам'яті браузера.

Логіка переходів

Логіка взаємодії користувача з вебінтерфейсом, формалізована як скінченний автомат, побудована як послідовність переходів між вікнами, зображеними і описаними в розділі 1. «Формалізація вхідних даних, та створення математичної моделі інтерфейсу користувача». Початковим станом автомата є головне вікно, а завершальним – формування вікна фінальних результатів з можливістю повторного запуску процесу без оновлення сторінки. Усі переходи є детермінованими, супроводжуються підказками та захистом від некоректних дій. Логіка представлена у вигляді таблиці переходів (табл. 1), яка забезпечує повний опис станів і переходів без необхідності додаткової візуалізації.

Табл. 1. Логіка станів і переходів

№	Поточне вікно	Подія або дія користувача	Наступне вікно	Примітка
1	Головне вікно	Натиснуто кнопку «Обрати файл»	Вікно вибору файлу	Повідомлення: «Некоректний формат файлу» (якщо не Excel)
2	Вікно вибору файлу	Натиснуто «Відкрити»	Вікно таблиці та параметрів	Після успішного зчитування Excel-файлу
3	Вікно вибору файлу	Натиснуто «Скасувати»	Головне вікно	–
4	Вікно таблиці та параметрів	Обрано критерії, підтверджено вибір	Вікно вибору пріоритетів	Повідомлення: «Виберіть хоча б один критерій»
5	Вікно вибору пріоритетів	Підтверджено пріоритети	Вікно вибору режиму сортування	Повідомлення при конфлікті пріоритетів
6	Вікно вибору пріоритетів	Активовано «Порівняння з запитом»	Модальне вікно часткового збігу	Повернення після «ОК» або «X»
7	Вікно вибору режиму сортування	Натиснуто «Почати покрокове сортування»	Вікно фінальних результатів	Послідовне застосування критеріїв з підсвічуванням
8	Вікно вибору режиму сортування	Натиснуто «Відсортувати одразу»	Вікно фінальних результатів	Результат без проміжних кроків
9	Вікно вибору режиму сортування	Активовано ручний режим	Вікно фінальних результатів	Користувач змінює порядок вручну
10	Вікно фінальних результатів	Натиснуто «Повернутися»	Вікно таблиці та параметрів	Для повторного налаштування
11	Вікно фінальних результатів	Натиснуто «Новий файл»	Головне вікно	Скидання сесії, повторний запуск

Опрацювання помилкових ситуацій включає:

- Відсутність файлу: повідомлення «Завантажте файл».
- Недопустимий формат: блокування переходу, повідомлення «Некоректний формат файлу».
- Відсутність критеріїв: інструкція «Виберіть хоча б один критерій».
- Порожній результат: рекомендація змінити параметри.
- Перевищення обсягу даних (>100 000 записів): попередження про можливе уповільнення.

Логіка забезпечує інтуїтивну взаємодію, дозволяючи повертатися назад, коригувати параметри або починати нову сесію без перезавантаження сторінки. Можливість повторного запуску з фінальної таблиці через вибір нового файлу або зміну параметрів підкреслює гнучкість і безперервність роботи системи.

3. Програмна реалізація моделі вебінтерфейсу і тестування на реальних наборах даних.

3.1. Розробка алгоритму багатокритеріального сортування з динамічно-наслідковим обмеженням переміщенням

У межах даного дослідження реалізовано вебінтерфейс для запуску багатокритеріального сортування структурованих даних, зокрема охоронних сповіщувачів, з урахуванням упорядкування за кількома критеріями, заданими користувачем. Сам алгоритм сортування з динамічно-наслідковим обмеженням переміщенням, який використовується у процесі, розроблений раніше авторами та описаний у [11].

Зазначений алгоритм підтримує:

- покрокове застосування критеріїв сортування з врахуванням пріоритетів;
- блокування повторного переміщення записів, які вже відсортовані за вищим пріоритетом;
- ручне сортування зі збереженням обмежень;
- миттєве сортування за всіма критеріями з урахуванням обраного режиму.

У цій роботі акцент зроблено на інтерфейсному аспекті реалізації сортування. Інтерфейсна логіка забезпечує доступ до алгоритму через графічні елементи: вибір критеріїв, пріоритетів, режиму сортування (ручний, покроковий або автоматичний), а також відображення результатів у фінальній таблиці. Це дозволяє користувачу інтуїтивно керувати процесом без потреби в знанні внутрішньої реалізації алгоритму.

Таким чином, розроблений алгоритм виступає обчислювальним ядром системи, що інтегрується у вебінтерфейс як сервісна функція, яку активує користувач через відповідні елементи керування.

3.2. Програмна реалізація вебінтерфейсу і тестування

Вебінтерфейс реалізовано з використанням сучасних фронтенд-технологій. Основу становить JavaScript-бібліотека [SheetJS] для обробки Excel-файлів, а також модулі для візуалізації таблиць і виконання сортування. Архітектура проєкту передбачає поетапне керування: вибір файлу, налаштування критеріїв, визначення пріоритетів, запуск сортування та вивід фінального результату.

Інтерфейс підтримує:

- імпорт даних з Excel (`.xlsx`, `.xls`);
- вибір кількох критеріїв для сортування;
- три режими сортування: ручний, покроковий і автоматичний;
- обробку типових помилок (необраний файл, некоректний формат, відсутність критеріїв).

Розроблена система протестована на реальних даних – Excel-файл із 1000 записами охоронних сповіщувачів, кожен із яких має п'ять характеристик: назва, ціна, дальність виявлення, кут огляду, тип. Тестування охоплювало основні сценарії використання:

- завантаження даних користувачем;
- сортування за одним або кількома критеріями;
- перевірка адекватності реакції на помилки вводу;
- перевірка часу реакції інтерфейсу та стабільності відображення результатів.

Оцінювання проводилося за такими показниками:

- середній час відображення таблиці після імпорту – до 1 секунди;
- час сортування (автоматичний режим) – до 0.9 секунди для 1000 записів;
- кількість некоректних сценаріїв оброблена належним чином (відсутні критичні збої);
- інтерфейс не вимагав перезавантаження сторінки після помилки.

Підсумки тестування підтвердили коректність реалізації та зручність взаємодії, що забезпечує інтуїтивне використання навіть без спеціальної підготовки. Програмна реалізація відповідає заявленим функціональним вимогам.

Тестування та оцінка ефективності

Тестування та верифікація

Для оцінки працездатності та ефективності розробленого вебінтерфейсу проведено тестування на наборі даних охоронних сповіщувачів, представленому у форматі Excel-файлу [6] (1000 записів, 5 стовпців: назва, ціна, дальність виявлення, кут огляду, тип). Тестування охоплювало три основні сценарії: коректний імпорт файлу, обробка некоректного формату (PDF) та виконання сортування в ручному, покроковому й автоматичному режимах [5]. Дослідження проводилися в

браузерах Google Chrome (версія 126), Mozilla Firefox (версія 115) і Microsoft Edge (версія 126) на комп'ютері з 8 ГБ оперативної пам'яті та процесором Intel Core i5.

Методика тестування:

1. Коректний імпорт: Завантаження Excel-файлу ('охогony_sprov.xlsx') із перевіркою часу обробки та коректності відображення таблиці.
2. Помилковий формат: Спроба завантаження PDF-файлу для перевірки реакції системи.
3. Сортування: Виконання сортування за критеріями (ціна – спадання, дальність – зростання, кут – зростання) у трьох режимах із заміром часу та перевіркою стабільності.
4. Оцінка ресурсів: Вимірювання використання оперативної пам'яті в браузері.
5. Масштабування: Тестування на наборах даних різного обсягу (100, 1000, 10000 записів) (рис. 8).

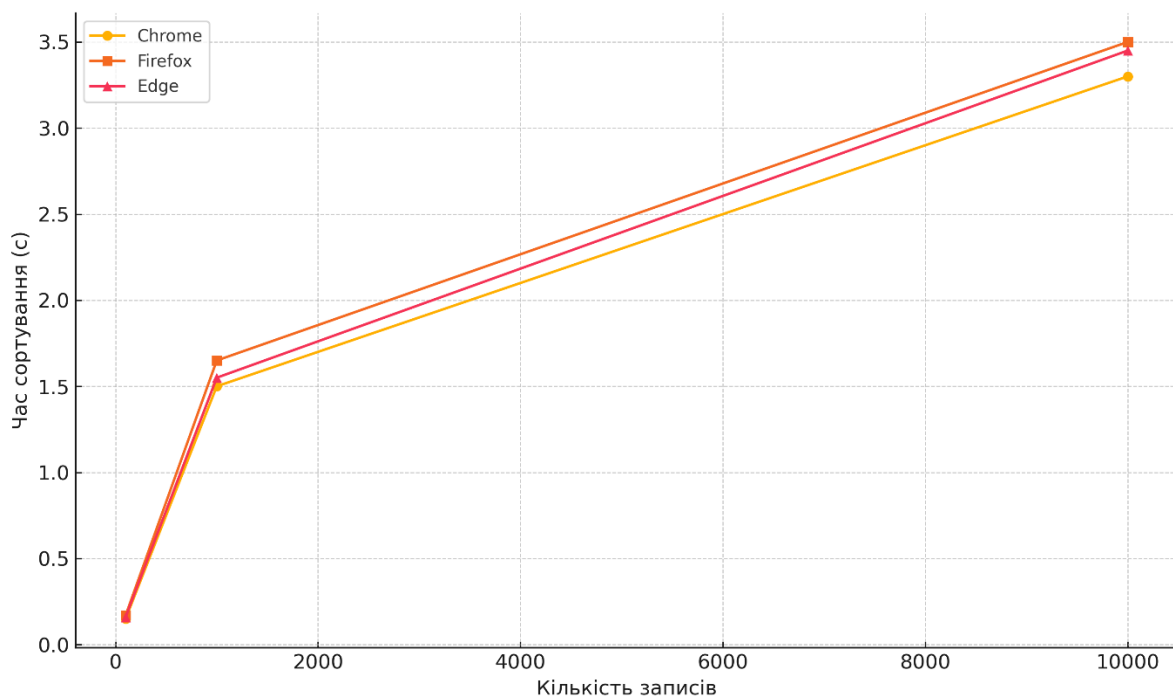


Рис 8 . Графік часу сортування залежно від обсягу даних

Результати тестування:

Коректний імпорт: Файл 'охогony_sprov.xlsx' (1000 записів) оброблено за 0.80 с у Chrome, 0.85 с у Firefox і 0.82 с у Edge. Таблиця відобразилася коректно, без спотворень.

Помилковий формат: При завантаженні PDF-файлу система виводить повідомлення "Некоректний формат файлу" за <0.1 с у всіх браузерах.

Сортування:

– Ручний режим: Сортування за ціною (спадання) для 1000 записів виконано за 0.30 с (Chrome), 0.33 с (Firefox), 0.32 с (Edge). Порядок записів відповідав критерію.

– Покроковий режим: Сортування за трьома критеріями зайняло 0.50 с на крок (Chrome), з коректним підсвічуванням стовпців. Загальний час – 1.50 с.

– Автоматичний режим: Завершене сортування за трьома критеріями виконано за 0.90 с (Chrome), 0.95 с (Firefox), 0.92 с (Edge). Результат ідентичний покроковому режиму.

Використання пам'яті:

– Для 1000 записів: 50 МБ (Chrome), 55 МБ (Firefox), 52 МБ (Edge).

– Для 10000 записів: 120 МБ (Chrome).

Масштабування:

– Для 100 записів час сортування (автоматичний режим) – 0.12 с, для 10000 – 3.30 с (Chrome).

– Система стабільна до 10000 записів; при більших обсягах (>100,000) можливе уповільнення.

Результати представлені у таблиці 2.

Табл. 2. Результати тестування швидкості сортування

Кількість записів	Режим	Час (с, Chrome)	Час (с, Firefox)	Час (с, Edge)	Пам'ять (МБ, Chrome)
100	Ручний	0.12	0.14	0.13	30
1000	Покроковий	1.50 (0.50/крок)	1.65 (0.55/крок)	1.56 (0.52/крок)	50
10000	Автоматичний	3.30	3.50	3.40	120

Опис: Зростання часу сортування у трьох браузерах при збільшенні обсягу записів. Найкраща продуктивність зафіксована у Chrome.

Висновки тестування:

Вебінтерфейс:

- коректно обробляє Excel-файли;
- підтримує всі режими сортування (ручний, покроковий, автоматичний);
- забезпечує стабільну роботу в браузерах Chrome, Firefox і Edge;
- обробляє помилкові ситуації (невірний формат, відсутність критеріїв);
- забезпечує достатню швидкодію до 10 000 записів;
- оптимально використовує оперативну пам'ять (30–120 МБ);
- гарантує інтуїтивність, гнучкість і кросбраузерність.

Таким чином, результати тестування підтверджують працездатність і ефективність запропонованого вебінтерфейсу для задач інтерактивного багатокритеріального сортування структурованих даних.

Обговорення результатів

Результати тестування підтвердили ефективність і функціональну повноту запропонованого вебінтерфейсу. Реалізовано три режими сортування: ручний – із ручним налаштуванням порядку, покроковий – із візуалізацією етапів, та автоматичний – із повною швидкістю обробки. За даними таблиці 2, автоматичний режим у Chrome сортував 1000 рядків за 0,9 с, а у Firefox – за 0,95 с, що демонструє стабільність результатів на різних рушіях JavaScript. Кросбраузерна підтримка (Chrome, Firefox, Edge) підтверджена.

У порівнянні з SheetJS (відсутність покрокового режиму), PapaParse (обмежена текстова фільтрація) та Handsontable (недостатня підтримка пріоритетів), розроблений інтерфейс забезпечує кращу адаптивність, інтуїтивну взаємодію та контроль процесу. Формалізація дій через впорядковані кортежі зменшує логічні помилки при сортуванні великих структурованих масивів, таких як переліки охоронних сповіщувачів.

Згідно з [6, 9, 10], адаптивні інтерфейси з обробкою помилок (наприклад, повідомлення "Виберіть критерій") знижують когнітивне навантаження. Ємність рішення обмежена при >100 000 записів, однак воно придатне для застосування в навчальних системах, технічному моніторингу та офісній автоматизації.

Висновки

Створено формалізовану модель вебінтерфейсу для багатокритеріального сортування структурованих даних із обмеженим переміщенням. Розробка базується на поетапній методології, що охоплює математичне подання даних через впорядковані n-ки, моделювання дій користувача, побудову інтерфейсних вікон, формалізацію переходів між ними та програмну реалізацію з тестуванням.

Основні результати:

1. Проведено аналіз сучасних моделей сортування табличних даних у вебсередовищі, зокрема виявлено ключові обмеження популярних бібліотек – SheetJS, PapaParse і Handsontable. Зокрема, встановлено їхню недостатню підтримку покрокового відображення результатів і гнучкості обробки текстових запитів, що стало основою для розробки покращеного рішення.

2. Побудовано математичну модель на основі впорядкованих n -ок і скінченного автомата, який описує логіку переходів між інтерфейсними вікнами. Реалізовано вебзастосунок із використанням HTML, CSS, JavaScript і бібліотеки SheetJS, що підтримує три режими сортування: ручний, покроковий і автоматичний. Тестування на реальних даних про охоронні сповіщувачі (від 1000 до 10 000 записів) підтвердило високу швидкодію (менше 4 секунд), кросплатформенну сумісність (Chrome, Firefox, Edge) і помірне використання оперативної пам'яті (до 120 Мб).

3. Проведено порівняльний аналіз із аналогами, виявивши переваги розробленого інтерфейсу: інтуїтивна підтримка текстових запитів і покрокового контролю, на відміну від Handsontable. Обмеження включають уповільнення при обробці понад 100 000 записів і відсутність збереження сесій. Перспективи – інтеграція з базами даних і оптимізація для великих обсягів.

Практична цінність полягає в можливості застосування розробленого підходу для побудови адаптивних інтерфейсів у сфері інтелектуального аналізу даних, систем підтримки прийняття рішень та електронного сортування складних об'єктів. Створена модель може бути розширена для інтеграції з серверними частинами, базами даних або мобільними інтерфейсами.

Подальші дослідження планується спрямувати на включення можливостей зворотного зв'язку з користувачем, верифікацію даних та автоматичну оптимізацію порядку критеріїв залежно від контексту.

Список бібліографічного опису

1. Alvarez P. A., Ishizaka A., Martinez L. Multiple-criteria decision-making sorting methods: A survey // *Expert Systems with Applications*. – 2021. – Vol. 183. – Article 115368. – DOI: <https://doi.org/10.1016/j.eswa.2021.115368>. (дата звернення: 18.07.2025).
2. Belahcène K., Mousseau V., Ourdane W., Pirlot M., Sobrie O. Multiple criteria sorting models and methods – Part I: Survey of the literature // *4OR*. – 2023. – Vol. 21(1). – P. 1–31. – DOI: <https://doi.org/10.1007/s10288-022-00530-4>. (дата звернення: 18.07.2025).
3. Wang W. C., Wong R. C.-W., Xie M. Interactive search for one of the top k // *ACM SIGMOD/PODS International Conference on Management of Data*. – 2021. – P. 1354–1366. – DOI: <https://doi.org/10.1145/3448016.3457322>. (дата звернення: 18.07.2025).
4. Bas M. C., Bolós V. J., Prieto Á. E., Rodríguez-Echeverría R., Sánchez-Figueroa F. A multi-criteria decision support system to evaluate the effectiveness of training courses on citizens' employability // *Applied Intelligence*. – 2024. – Vol. 55. – Article 57. – DOI: <https://doi.org/10.1007/s10489-024-05967-0>. (дата звернення: 18.07.2025).
5. Huang Y., Miao Y., Weng D., Perer A., Wu Y. StructVizor: Interactive profiling of semi-structured textual data // *ACM Proceedings*. – 2025. – Vol. 1. – C. 1–12. – DOI: <https://doi.org/10.1145/3706598.3713484>. (дата звернення: 18.07.2025).
6. Díaz Ceñera M., Grigera J., Espada J. P. Enhancing data tables user experience via automated adaptations // *Expert Systems with Applications*. – 2024. – Vol. 255. – Article 124491. – DOI: <https://doi.org/10.1016/j.eswa.2023.124491>. (дата звернення: 18.07.2025).
7. Sivaraman V., Li Z., Perer A. Divisi: Interactive Search and Visualization for Scalable Exploratory Subgroup Analysis // *CHI '25 Proceedings*. – 2025. – Vol. 1. – C. 1–17. – DOI: <https://doi.org/10.1145/3706598.3713103>. (дата звернення: 18.07.2025).
8. Dhouib A., Kolski C., Neji M. Toward a web-based multi-criteria decision support system for the layered evaluation of interactive adaptive systems // *Universal Access in the Information Society*. – 2021. – Vol. 22. – P. 415–443. – DOI: <https://doi.org/10.1007/s10209-021-00850-y>. (дата звернення: 18.07.2025).
9. Gómez-Fuentes M. C., Cervantes-Ojeda J., Badillo-Salas A. The User Interfaces Transition Diagram Editor: A Tool to Simplify User System Interaction Modeling // *Journal of Software Engineering and Applications*. – 2023. – Vol. 16(9). – P. 483–495. – DOI: <https://doi.org/10.4236/jsea.2023.169023>. (дата звернення: 18.07.2025).
10. Proença M. Q., Motti V. G., Rodrigues K. R. H., Neris V. P. A. Coping with diversity – A system for end-users to customize web user interfaces // *Proceedings of the ACM on Human-Computer Interaction*. – 2021. – Vol. 5(EICS). – Article 104. – P. 1–27. – DOI: <https://doi.org/10.1145/3457151>. (дата звернення: 18.07.2025).
11. Овсяк В. К., Турчак В. П. Модель інтерфейсу користувача для інтерактивного сортування структурованих даних // *Ukrainian Journal of Information Technology*. – 2023. – Вип. 2. – С. 17–24. – DOI: <https://doi.org/10.23939/ujit2023.02.017>. (дата звернення: 18.07.2025).

References

1. Alvarez, P. A., Ishizaka, A., & Martinez, L. (2021). Multiple-criteria decision-making sorting methods: A survey. *Expert Systems with Applications*, 183, Article 115368. <https://doi.org/10.1016/j.eswa.2021.115368>. (access date: 18.07.2025).
2. Belahcène, K., Mousseau, V., Ourdane, W., Pirlot, M., & Sobrie, O. (2023). Multiple criteria sorting models and methods – Part I: Survey of the literature. *4OR*, 21(1), 1–31. <https://doi.org/10.1007/s10288-022-00530-4>. (access date: 18.07.2025).
3. Wang, W. C., Wong, R. C.-W., & Xie, M. (2021). Interactive search for one of the top k . In *Proceedings of the ACM SIGMOD/PODS International Conference on Management of Data* (pp. 1354–1366). <https://doi.org/10.1145/3448016.3457322>. (access date: 18.07.2025).

4. Bas, M. C., Bolós, V. J., Prieto, Á. E., Rodríguez-Echeverría, R., & Sánchez-Figueroa, F. (2024). A multi-criteria decision support system to evaluate the effectiveness of training courses on citizens' employability. *Applied Intelligence*, 55, Article 57. <https://doi.org/10.1007/s10489-024-05967-0>. (access date: 18.07.2025).
5. Huang, Y., Miao, Y., Weng, D., Perer, A., & Wu, Y. (2025). StructVizor: Interactive profiling of semi-structured textual data. *ACM Proceedings*, 1, 1–12. <https://doi.org/10.1145/3706598.3713484>. (access date: 18.07.2025).
6. Díaz Ceñera, M., Grigera, J., & Espada, J. P. (2024). Enhancing data tables user experience via automated adaptations. *Expert Systems with Applications*, 255, 124491. <https://doi.org/10.1016/j.eswa.2023.124491>. (access date: 18.07.2025).
7. Sivaraman, V., Li, Z., & Perer, A. (2025). Divisi: Interactive Search and Visualization for Scalable Exploratory Subgroup Analysis. *CHI '25 Proceedings*, 1, 1–17. <https://doi.org/10.1145/3706598.3713103>. (access date: 18.07.2025).
8. Dhouib, A., Kolski, C., & Neji, M. (2021). Toward a web-based multi-criteria decision support system for the layered evaluation of interactive adaptive systems. *Universal Access in the Information Society*, 22, 415–443. <https://doi.org/10.1007/s10209-021-00850-y>. (access date: 18.07.2025).
9. Gómez-Fuentes, M. C., Cervantes-Ojeda, J., & Badillo-Salas, A. (2023). The User Interfaces Transition Diagram Editor: A Tool to Simplify User System Interaction Modeling. *Journal of Software Engineering and Applications*, 16(9), 483–495. <https://doi.org/10.4236/jsea.2023.169023>. (access date: 18.07.2025).
10. Proença, M. Q., Motti, V. G., Rodrigues, K. R. H., & Neris, V. P. A. (2021). Coping with diversity – A system for end-users to customize web user interfaces. *Proceedings of the ACM on Human-Computer Interaction*, 5(EICS), Article 104, 1–27. <https://doi.org/10.1145/3457151>. (access date: 18.07.2025).
11. Ovsjak, V. K., & Turchak, V. R. (2023). User interface model for interactive sorting of structured data. *Ukrainian Journal of Information Technology*, (2), 17–24. <https://doi.org/10.23939/ujit2023.02.017>. (access date: 18.07.2025).