УДК 004

# ВИКОРИСТАННЯ ФРЕЙМВОРКУ CHARTS.JS ПРИ ПОБУДОВІ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ КОНТРОЛЮ ЯКОСТІ ПРОДУКЦІЇ

**О. Д. Скоробреха,**
студент 4 курсу заочної форми навчання
спеціальності «Комп'ютерні науки»
**С.В. Ковбасюк**, д.т.н., с.н.с.

**Постановка проблеми та аналіз останніх досліджень:** Системи побудови графіків та діаграм набули широкого використання в багатьох сферах діяльності людства. Реалізація подібного функціоналу не є стандартним інструментом популярних мов програмування, які залучаються для створення інформаційних систем, тому виконання задач з відображення графіків спирається на сукупність нестандартних бібліотек, які реалізують такі можливості.

**Основний матеріал:** Функціонал побудови графіків та діаграм не представлений у стандартному наборі інструментів веб – розробника, тому реалізація таких функцій лягає на створення сукупності елементів з подальшим заведенням їх до сукупного тегу – canvas, який клієнт користувача відмальовує завдяки сучасному функціоналу веб – переглядачів. Canvas API надає можливості для малювання графіки через JavaScript і HTML <canvas> – елемент. Крім усього іншого, його можна використовувати для анімації, ігрової графіки, візуалізації даних, маніпулювання зображеннями та обробки відео в режимі реального часу[3].

В основі роботи API для побудови графіків лежить методологія ООП, де процес реалізації окремих елементів системи спирається на створення окремих об'єктів з унікальними полями, які можуть приймати різні значення для кастомізації відображуваного контенту користувачу. При початку роботи з бібліотекою є необхідність задати першочергові параметри, які впливають на представлення графічного контенту оператору, до них відносяться поля:

- Назва графіку чи діаграми;
- Тип побудови графічного контенту;
- Стиль ліній;
- Розмір;
- Початок відліку;

Після задавання першочергових параметрів необхідним кроком є – створення колекції вхідних даних для подальшого представлення їх користувачу. Для цього доцільно буде використати масиви інформаційних змінних, у який зберігаються значення кількісних або якісних складових наявної продукції. Надалі процес побудови графіків зводиться до виклику стандартних методів API з передаванням їм у якості аргументів колекцій даних, які необхідно відобразити графічно. Методика отримання даних спирається на методи getElementById та getElementsByClassName [2] при селекції окремих елементів сторінки. Суть вибірки полягає у зверненні до вмісту окремих тегів за їх унікальними ідентифікаторами або класами, це дає змогу розділяти та групувати подібну інформацію. Для реалізації адаптивності вмісту слід використовувати медіа – запити, результат виконання яких залежить від роздільної здатності клієнта оператора, тому в залежності від відповіді формується набір параметрів каскадної таблиці стилів, які визначають масштаб та дизайн відображуваних елементів користувачу. При розробці подібного функціоналу слід звернутися до документації останніх версій веб – переглядачів, в яких описано функціонал та перелік підтримуваних технологій, слід звернути увагу на те, які теги та функціональні елементи підтримує сам браузер, адже це напряму впливає на вигляд представлення окремих елементів користувачу. Для внесення розмітки на графік доцільно

використовувати параметри мінімального та максимального значення відображуваної інформації, це можна реалізувати завдяки алгоритмам сортування даних, де в результаті виконання методу дані представляються у порядку зростання чи спадання, а шукані елементи знаходяться на початку та вкінці ряду даних[1]. Після отримання значень слід знайти абсолютне значення, яке в подальшому слід розділити на кількість відображуваних елементів, а після того адаптивно відобразити лінії, виходячи з результату виконання медіа – запиту

**Висновки:** Бібліотека Charts.js надає широкий спектр функцій для побудови та відображення графіків і діаграм. Функціонал фреймворку спирається на використання сучасних веб-елементів та тегів, які керуються за допомогою скриптів на мові Java Script. При тестуванні на швидкодію та адаптивність система, яка створена при залучені вище вказаної бібліотеки показала, що швидкодія відображення графіків користувачу не знижує ефективність завантаження сторінки і не навантажує трафік користувача.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chart.js, керівництво користувача URL: https://www.chartjs.org/docs/. Електронний ресурс.

2. JavaScript Tutorial. Examples in Each Chapter URL: https://www.w3schools.com/js/ Електронний ресурс.

3. Canvas API URL: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API?retiredLocale=uk Електронний ресурс.