

УДК 004.4

Буряк Д., Мудрик І.

Тернопільський національний технічний університет імені Івана Пулюя

ПІДХОДИ ДО РЕАЛІЗАЦІЇ ФУНКЦІОНАЛУ ПЕРЕВІРКИ ТЕКСТУ У БРАУЗЕРНОМУ СЕРЕДОВИЩІ

Buriak D., Mudryk I.

Ternopil Ivan Puluj National Technical University

APPROACHES TO IMPLEMENTING TEXT VERIFICATION FUNCTIONALITY IN THE BROWSER ENVIRONMENT

У наш час текстова інформація є одним із ключових засобів комунікації в цифровому просторі. Щоденно мільйони користувачів взаємодіють із веб-застосунками, створюючи текстовий контент: публікації, повідомлення, електронні листи, коментарі тощо. Проте, незважаючи на широкий спектр платформ, далеко не всі вони мають вбудовані інструменти для перевірки орфографії, граматики чи стилістики. Як результат — зниження якості контенту, втрати довіри до ресурсу або навіть викривлення змісту через незначні помилки. Ця проблема є особливо актуальною в умовах зростання кількості онлайн-комунікацій та необхідності дотримання мовних стандартів. Відсутність перевірки тексту може також негативно впливати на професійний імідж як окремих користувачів, так і компаній загалом.

Одним з ефективних рішень цієї проблеми є розробка розширень для браузера, які б дозволяли автоматично перевіряти та виправляти текст безпосередньо в інтерфейсі користувача. Такий підхід забезпечує універсальність, оскільки дозволяє інтегруватися з будь-яким вебсайтом, не змінюючи його вихідний код. Основна мета проєкту полягає у створенні функціонального, гнучкого та ефективного розширення, здатного аналізувати введений текст у режимі реального часу, виявляти помилки та пропонувати користувачу можливі варіанти виправлення. Такий інструмент може суттєво підвищити зручність написання тексту, особливо у середовищах, де відсутні вбудовані системи перевірки.[3][4]

Важливу роль у реалізації проєкту відіграє технологія Node.js — середовище виконання JavaScript-коду на стороні серверу. У межах проєкту Node.js використовується для створення серверної частини, яка відповідає за обробку тексту, взаємодію з зовнішніми API (наприклад, сервісами перевірки граматики) та формування відповіді для клієнта. Завдяки асинхронній моделі Node.js забезпечується швидка обробка запитів, що є критичним для забезпечення роботи розширення в реальному часі. Крім того, використання спільної мови програмування для клієнтської та серверної частин (JavaScript) спрощує розробку і підтримку проєкту.[2]

Для побудови інтерфейсу розширення використовується бібліотека React. Вона дозволяє створити динамічний UI, який адаптується під дії користувача. React забезпечує зручну роботу з компонентами, станом додатка та ефективне оновлення інтерфейсу, що робить його ідеальним для інтеграції з браузерним середовищем, зокрема, для створення інтерактивного вікна з підказками, налаштуваннями, підсвіткою помилок тощо. Такий підхід дозволяє реалізувати зручну та інтуїтивно зрозумілу взаємодію з користувачем навіть у складних сценаріях перевірки тексту.[1]

Під час реалізації розширення також враховано питання безпеки, продуктивності та зручності використання. Завдяки поєднанню Node.js і React є

можливість досягти балансу між продуктивністю на сервері та зручністю користувацького інтерфейсу.

Для ефективної перевірки тексту сучасні розширення браузера активно використовують зовнішні API, зокрема ті, що базуються на технологіях штучного інтелекту. Саме така інтеграція API надає можливість проводити глибокий аналіз тексту з урахуванням граматики, пунктуації, стилістики й навіть контексту вживання слів. Завдяки цьому перевірка стає не лише формальною, а й змістовною, що особливо важливо при створенні складних або професійних текстів. Інтеграція подібних сервісів у розширення здійснюється через HTTP-запити, а результати аналізу повертаються у структурованому форматі (наприклад, JSON), що спрощує їх обробку на клієнтській стороні.

Список використаних джерел:

1. Cases. Що таке React JS? Як почати вивчати Реакт? Навички для React Developer: <https://cases.media/article/sho-take-react-js-yak-pochati-vivchati-reakt-navichki-dlya-react-developer?srsltid=AfmBOoq-mjrXdyqqgfB5sVg65Pl0ZEwNRb73oOKQlp3mSaQqjMYSb5YP>
2. Node.js. About Node.js: <https://nodejs.org/en/about>
3. Towardsdatascience. Grammar Error Correction: <https://towardsdatascience.com/grammar-error-correction-af365dad794/>
4. Machine learning models and methods aspects of processing unstructured data. Bryk, I Mudryk, M Holubovskyi, Y Stoianov. (BAIT 2024) Zboriv, Ukraine, 2024. pp. 64-74: <https://ceur-ws.org/Vol-3842/paper4.pdf>
5. Awe Amason. What is an API (Application Programming Interface)?: <https://aws.amazon.com/what-is/api/>