

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

_____ червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна система каталогізації інструментів веб-розробника»
здобувача групи ІН – 94 – 1 Якименка Ярослава Олександровича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

_____ Ярослав ЯКИМЕНКО
(підпис)

Керівник,
професор, доктор технічних наук

Микола БУДНІК

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
 здобувача групи ІН-94-1 Якименка Ярослава Олександровича

1. Тема роботи: «Інформаційна система каталогізації інструментів веб-розробника»
затверджую наказом по СумДУ від _____
2. Термін здачі здобувачем кваліфікаційної роботи _____
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
 - 1) *Аналіз проблеми предметної області, постановка й формування завдань дослідження.*
 - 2) *Огляд технологій, що використовуються у інформаційних системах каталогізації інструментів веб-розробника.*
 - 3) *Розробка інформаційної системи каталогізації інструментів веб-розробника.*
 - 4) *Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для інструментів веб-розробника</i>		
3	<i>Розробка інформаційної системи каталогізації інструментів</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 61 стр., 54 рис., 1 додаток, 22 джерела.

Обґрунтування актуальності теми роботи – Зростаючий обсяг інформації та швидкий розвиток технологій веб-розробки ставлять перед розробниками веб-сайтів складні завдання. Необхідність у швидкому доступі до актуальних інструментів та систематизації цих інструментів стає критично важливою. Тема кваліфікаційної роботи має на меті створення інформаційної системи каталогізації інструментів веб-розробника, що спростить вибір та пошук інструментів, підвищить продуктивність та якість роботи розробників.

Об'єкт дослідження — каталогізація інструментів веб-розробника.

Мета роботи — реалізація інформаційної системи каталогізації інструментів веб-розробника, яка полегшить вибір та пошук інструментів, покращить ефективність роботи та зменшить витрати часу на рутинні задачі.

Методи дослідження — аналіз потреб веб-розробників, проектування архітектури системи, програмування системи та тестування її роботи.

Результати — розроблено інформаційну систему каталогізації інструментів веб-розробника, яка забезпечує швидкий доступ до актуальних інструментів, полегшує вибір та пошук, покращує ефективність та якість роботи розробників.

ІНФОРМАЦІЙНА СИСТЕМА, ІНСТРУМЕНТИ ВЕБ-РОЗРОБНИКА,
КАТАЛОГІЗАЦІЯ, ПРОДУКТИВНІСТЬ, РНР

ЗМІСТ

ВСТУП	5
1 АНАЛІТИЧНИЙ ОГЛЯД	8
1.1 Дослідження сучасного стану	8
1.2 Критерії аналізу та визначення проблем	10
1.3 Аналіз аналогічних проектів	12
1.4 Постановка задачі	15
2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ	17
2.1 Інформаційна модель системи каталогізації інструментів	17
2.2 Прототипування веб-сайту	19
2.3 Інструменти, мови програмування, фреймворки	20
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	26
3.1 Проектування інформаційної системи	26
3.2 Програмна реалізація	29
3.3 Тестування інструментів інформаційної системи	37
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ	58
Додаток А. Лістинг програмного коду	58

ВСТУП

Актуальність. У сучасному світі інформаційних технологій швидкість розвитку та зростання обсягу інформації веб-розробки є вражаючою. Із появою нових технологій, фреймворків та інструментів, веб-розробники стикаються з викликами, пов'язаними з пошуком, вибором інструментів та їх використанням у роботі [1]. Висока складність та гетерогенність сучасного веб-оточення призводять до зайвої витрати часу на рутинні задачі, такі як знаходження відповідних бібліотек, фреймворків, систем управління та тестування.

Така ситуація вимагає ефективних методів каталогізації та систематизації інструментів веб-розробки. «Інформаційна система каталогізації інструментів веб-розробника» може відповісти на ці потреби, надаючи розробникам швидкий доступ до необхідних інструментів, полегшуючи вибір, забезпечуючи сумісність та підвищуючи якість роботи.

Забезпечення доступу до актуальних, релевантних інструментів є ключовим фактором для успішної веб-розробки. Система каталогізації інструментів веб-розробника може допомогти розробникам зосередитися на творчій роботі, зменшити витрати часу на пошук та вибір інструментів, а також підвищити продуктивність та якість виконаної роботи. Така система стає необхідним інструментом для сучасних веб-розробників, які прагнуть досягти успіху в конкурентному світі веб-розробки.

Об'єкт дослідження. Об'єктом дослідження є сама система, її функціональність, інтерфейс, архітектура та ефективність у роботі з реальними веб-розробниками. Метою дослідження є розробка та реалізація цієї системи з метою полегшення процесу вибору та пошуку інструментів для веб-розробників, покращення їхньої ефективності та зменшення витрат часу на рутинні задачі.

Предмет дослідження. Предметом дослідження є "Інформаційна система каталогізації інструментів веб-розробника". Основною метою дослідження є розробка та реалізація цієї системи, яка спрямована на полегшення процесу пошуку, вибору та використання інструментів для веб-розробників.

Предмет дослідження охоплює такі аспекти:

1) **Аналіз потреб веб-розробників.** В ході дослідження буде проведений аналіз запитів веб-розробників для вивчення їхніх потреб та вимог до інструментів. Це дозволить визначити ключові функціональні можливості, які має містити система.

2) **Проектування інформаційної системи.** Буде проведено проектування архітектури системи, розробка інтерфейсу та бази даних, а також визначення логічної структури системи. Це дозволить створити ефективну та зручну систему для веб-розробників.

3) **Реалізація і тестування системи.** Буде проведено програмування системи, її інтеграцію з іншими інструментами та технологіями, а також тестування для перевірки якості та функціональності. Це дозволить забезпечити ефективну роботу системи та її відповідність потребам веб-розробників.

Гіпотеза. Гіпотеза дослідження полягає в тому, що «Інформаційна система каталогізації інструментів веб-розробника» здатна полегшити процес вибору, пошуку та використання інструментів для веб-розробників, покращити їхню ефективність та якість роботи, а також зменшити витрати часу та ресурсів на рутинні задачі.

Гіпотеза базується на передбаченні, що розроблена інформаційна система надасть веб-розробникам зручний інструментарій для швидкого доступу до актуальних інструментів, а також забезпечить їхню сумісність та якість виконаної роботи. Це допоможе розробникам ефективно виконувати завдання, уникнути зайвих витрат часу та помилок, а також підвищити якість та продуктивність роботи [2].

Новизна. Новизна дослідження полягає в розробці та реалізації «Інформаційної системи каталогізації інструментів веб-розробника», яка вирішує актуальну проблему рутинних задач та ефективного вибору інструментів для веб-розробників. Однією з основних новаторських рис системи є об'єднання різноманітних інструментів веб-розробки під одним дахом, що

полегшує їхнє використання та сумісність. Також, система буде мати можливість оновлюватись та доповнюватись з новими інструментами та технологіями, що забезпечить її актуальність у динамічному світі веб-розробки.

Крім того, розроблена система буде враховувати потреби веб-розробників, забезпечуючи їм широкий вибір інструментів з урахуванням потреб та технічних вимог. Це дозволить забезпечити високу релевантність та корисність системи для веб-розробників.

Структура. Дане робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатку.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Дослідження сучасного стану

У світі веб-розробки, що стрімко розвивається, бути в курсі новітніх інструментів і технологій є вирішальним для досягнення успіху. Однак, однією з основних проблем, з якою стикаються веб-розробники, є відсутність єдиної інформаційної системи для каталогізації інструментів. Це призводить до значних витрат часу при виборі відповідних інструментів та фрагментарного досвіду пошуку інструментів з різних джерел [3]. Це дослідження має на меті дослідити актуальність цієї проблеми, висвітливши трудомісткість вибору інструментів, відсутність комплексного каталогу інструментів та їх важливість в оптимізації часу для веб-розробників в Україні та світі.

Процес вибору правильних інструментів для веб-розробки часто забирає багато сил і часу. Зважаючи на широкий спектр доступних варіантів, веб-розробники витрачають значну кількість часу на дослідження [4], порівняння та оцінку різних інструментів. Цей дорогоцінний час можна було використати для виконання реальних завдань розробки, але це призвело до пониження продуктивності та ефективності веб-розробників.

Згідно з опитуванням, проведеним незалежними дослідниками, приблизно 57% опитаних сучасних розробників повідомили [5], що витрачають кілька годин щотижня на пошук та оцінку інструментів та інформації. Це означає значну втрату продуктивного робочого часу, який можна було використати для кодування, налагодження або покращення існуючих проектів.

Ще одним аспектом, який загострює проблему, є відсутність централізованого каталогу інструментів для веб-розробки. Наразі розробники покладаються на різні веб-сайти, такі як онлайн-інструменти, сайти з навчальними матеріалами та соціальні медіа-платформи, щоб знайти нові інструменти для оптимізації розробки. Хоча ці джерела та існуючі веб-сайти, пропонують цінну інформацію та функціонал, відсутність консолідації ускладнює відстеження всіх доступних варіантів.

Крім того, оскільки кількість інструментів продовжує зростати, знайти найбільш підходящий інструмент для конкретного завдання стає все складніше. Розробникам часто доводиться переходити з одного веб-сайту на інший, просіювати застарілу інформацію та стикатися зі зламаними або неактивними інструментами. Такий розрізнений досвід не лише забирає дорогоцінний час, але й знижує загальну ефективність та результативність робочих процесів веб-розробки.

Щоб зрозуміти актуальність проблеми в Україні та світі, важливо проаналізувати статистику використання додаткових інструментів розробниками. Нещодавнє дослідження, проведене незалежною компанією GitHub, показало, що понад 95% веб-розробників в Україні та світі використовують додаткові інструменти [6], окрім свого основного середовища розробки для оптимізації робочих процесів.

Ці додаткові інструменти включають доповнення коду, його оптимізації, використання додаткових бібліотек та фреймворків, штучного інтелекту, тощо. Хоча ці інструменти є важливими для підвищення продуктивності та забезпечення ефективних робочих процесів, відсутність єдиної інформаційної системи для вирішення простих задач та завдань ускладнює для розробників пошук та використання найбільш підходящих інструментів для їх проектів.

У швидкоплинному світі веб-розробки час є цінним ресурсом. Здатність оптимізувати час, витрачений на другорядні завдання, такі як вибір інструментів, безпосередньо впливає на загальну ефективність і конкурентоспроможність веб-розробників. Єдина інформаційна система для каталогізації інструментів може суттєво сприяти оптимізації часу, надаючи централізовану платформу, яка пропонує багатий інструментарій, аналоги та порівняння різних інструментів.

Маючи доступ до такої системи, веб-розробники можуть швидко використати доступні інструменти, порівняти їх можливості та функціонал, прочитати відгуки користувачів і прийняти обґрунтоване рішення без необхідності проведення тривалих досліджень.

Крім того, уніфікована інформаційна система може сприяти співпраці та обміну знаннями всередині спільноти веб-розробників, бо вони зможуть ділитися своїм досвідом, рекомендаціями та найкращими практиками, пов'язаними з різними інструментами, сприяючи розвитку культури безперервного вдосконалення інструментів для спрощення своєї роботи.

Відсутність єдиної «Інформаційної системи каталогізації інструментів веб-розробника» є актуальною та значущою проблемою в сучасному цифровому світі. Значні витрати часу на вибір інструментів, відсутність консолідованого каталогу та їх важливість в оптимізації часу - підкреслюють необхідність комплексного вирішення цієї проблеми. Вирішивши цю проблему та забезпечивши централізовану платформу для пошуку та оцінки інструментів, веб-розробники в Україні та в усьому світі зможуть підвищити свою продуктивність, оптимізувати робочі процеси та залишатися конкурентоспроможними в індустрії, що постійно розвивається [7].

Важливо, щоб зацікавлені сторони, лідери галузі та самі розробники визнали важливість цієї проблеми та працювали над створенням інклюзивної та уніфікованої «Інформаційної системи каталогізації інструментів веб-розробника».

1.2 Критерії аналізу та визначення проблем

«Інформаційна система каталогізації інструментів веб-розробника» є важливим ресурсом для веб-розробників, що дозволяє їм знаходити, обирати та використовувати найбільш підходящі інструменти для своїх проектів. У цьому аналізі, ми порівнюємо різні аспекти такої системи, включаючи простоту використання, зручність, функціональність, адаптивність до різних пристроїв користувачів (смартфон, планшет або персональний комп'ютер), доступний функціонал і наявність реєстрації облікових записів.

Основні критерії аналізу та визначення проблем серед існуючих веб-сайтів, які надають інструментарій подібній до «Інформаційної системи»:

1) **Простота використання.** Простота використання «Інформаційної системи каталогізації інструментів веб-розробника» має першорядне значення для її ефективності. Розробникам потрібен зручний інтерфейс, який дозволить їм легко орієнтуватися в системі і швидко знаходити потрібну інформацію. Добре розроблена система повинна надавати інтуїтивно зрозумілі можливості пошуку та фільтрації, чітку категоризацію інструментів та стислі описи, які допомагають у процесі прийняття рішень.

2) **Зручність.** Зручність охоплює загальний досвід користувача та його задоволення від взаємодії з системою каталогізації. Зручна для користувача система, має надавати пріоритет ефективності, оперативності та доступності. Вона має бути оптимізована для забезпечення швидкого завантаження, зручної навігації та сумісності з різними браузерами і пристроями.

3) **Функціональність.** Функціональність «Інформаційної системи каталогізації інструментів веб-розробника» відіграє вирішальну роль у її корисності. Вона повинна надавати вичерпну та актуальну інформацію про різні інструменти, включаючи опис функціоналу. Крім того, система може надавати порівняння, рекомендації, засновані на алгоритмах, частотою використання інших інструментів та думках експертів в області веб-програмування.

4) **Адаптивність.** Здатність системи каталогізації адаптуватися до різних пристроїв, таких як смартфони, планшети або персональні комп'ютери, має важливе значення для кінцевого користувача, адже адаптивний дизайн, який підлаштовується під різні розміри та роздільну здатність екрану, гарантує доступ до інформаційної системи для більшості веб-розробників у світі. Така адаптивність гарантує, що розробники можуть отримати доступ до системи каталогізації незалежно від пристрою, яким вони користуються.

5) **Доступні інструменти.** Наявність широкого спектру функціональних можливостей в «Інформаційній системі каталогізації інструментів веб-розробника» є життєва важливою для задоволення різноманітних потреб. Вона повинна пропонувати розширені можливості пошуку, різноманітний функціонал

або наявність спеціалізованих інструментів для подальшого використання. Крім того, такі функції, як сортування інструментів за популярністю або схожістю, можуть ще більше підвищити зручність та ефективність системи, адже допоможе веб-розробникам у реалізації своїх проєктів через наявність потрібних у подальшому використанні інструментів.

б) **Доступність реєстрації користувача.** Доступність реєстрації є критично важливим аспектом «Інформаційної системи каталогізації інструментів веб-розробника». Добре розроблена система каталогізації, повинна забезпечувати простий, безпроблемний процес реєстрації, щоб заохочувати користувача брати участь у розвитку інформаційної системи і залучати до використання нових інструментів, які знаходяться на стадії розробки або тестування.

1.3 Аналіз аналогічних проєктів

Після огляду мережі Інтернет, були знайдені такі веб-сайти, що задовольняють вимоги подальшого аналізу: rapidtables.com, ipvoid.com, whatismyip.com. В результаті огляду веб-сайтів, були зроблені наступні висновки, складено порівняльну таблицю 1.1 та опис проєктів:

1) **Веб-сайт «rapidtables.com»** (рис. 1.1) пропонує широкий спектр онлайн-інструментів, серед яких можна знайти утиліти для роботи з текстом, конвертери одиниць вимірювання, генератори випадкових чисел, кодувальники, декодувальники та інше.

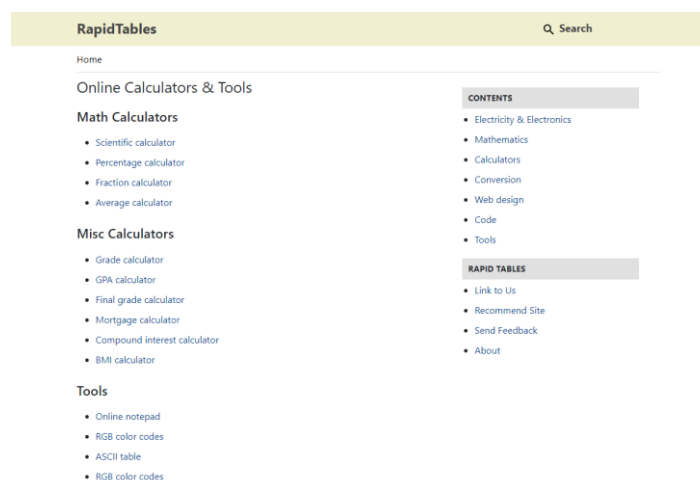


Рисунок 1.1 – Веб-сайт «rapidtables.com»

2) Веб-сайт «ipvoid.com» (рис. 1.2) є онлайн-інструментом, який дозволяє користувачам перевірити та дізнатись власника IP-адреси або домену, а також його доступність, мережеву інформацію, тощо.

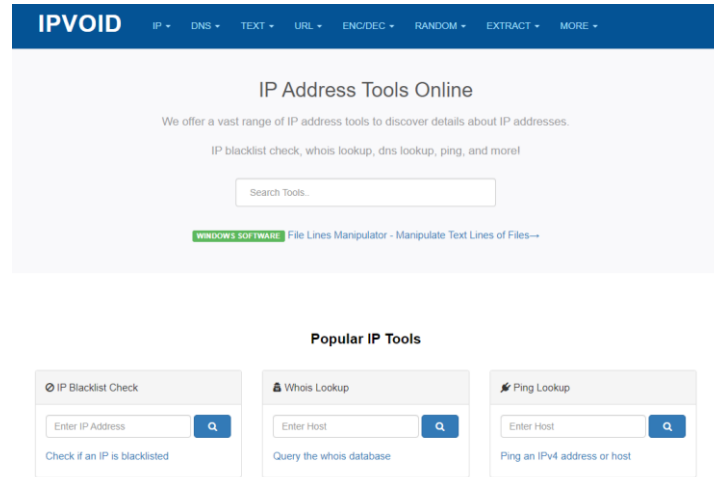


Рисунок 1.2 – Веб-сайт «ipvoid.com»

3) Веб-сайт «whatismyip.com» (рис. 1.3) є сервісом, який надає інформацію про вашу IP-адресу та інші мережеві дані. Ви можете дізнатися свою поточну IP-адресу, а також отримати інформацію про ваш браузер та операційну систему.

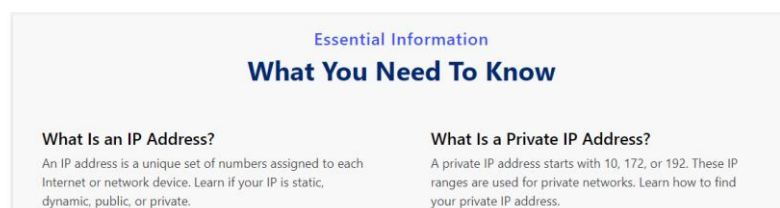
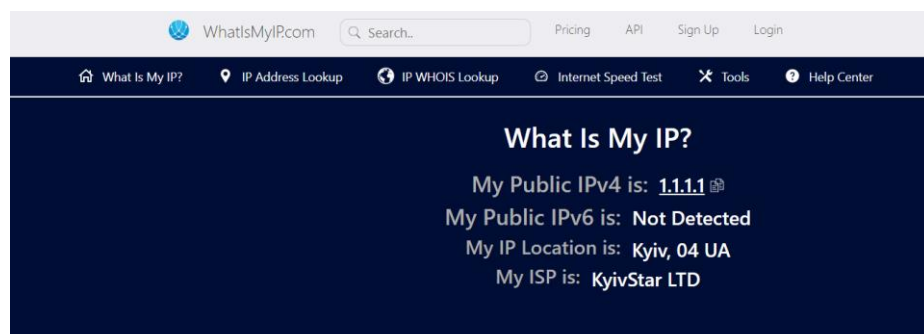


Рисунок 1.3 – Веб-сайт «whatismyip.com»

Таблиця 1.1 – Порівняльна характеристика веб-каталогів інструментів

	rapidtables.com	ipvoid.com	whatismyip.com
Простота використання (1-5)	3 (непростий для користувача через відсутність перекладу сайту)	5 (усі інструменти доступні на сайті одразу)	3 (непогано структурований, але мало інформації з головної сторінки)
Зручність (1-5)	1 (не адаптований до різних пристроїв, наприклад: смартфон, планшет)	5 (гарно зроблений, доступний на всіх пристроях)	2 (складно зроблений, доступний лише користувачам ПК)
Функціональність (1-5)	3 (деякі доступні інструменти, але не для веб-розробників)	5 (усі інструменти для веб-розробника)	4 (майже всі інструменти для веб-розробника)
Адаптивність (так, ні)	Ні, відсутня	Так, автоматична	Ні, доступний лише користувачам с ПК
Доступні інструменти (1-5)	1 (відсутність більшості інструментів для веб-розробки)	3 (відсутність частини інструментів для веб-розробки)	3 (відсутність частини інструментів для веб-розробки)
Реєстрація користувача (так, ні)	Ні, відсутня	Ні, відсутня	Так, присутня

Зважаючи на всі пункти порівняльної характеристики з таблиці 1.1, можна зробити наступні висновки по недолікам та функціональним можливостям:

- 1) більшість веб-сайтів мають середній бал за простотою використання, тобто вони будуть зрозумілі кожному користувачу, але через перевантаженість дизайну не будуть зрозумілі більшості;

- 2) більшість веб-сайтів мають погану зручність, тобто вони доступні для використання з будь-якого пристрою, але мають недоліки використання, наприклад зі смартфона або планшета;
- 3) функціональні можливості більшості з представлених веб-сайтів або недостатні, або не задовольняє потреби веб-розробників у необхідних інструментах через їх фізичну відсутність;
- 4) адаптивність у більшості веб-сайтів або відсутня, або доступна лише у декількох випадках, наприклад: веб-сайт доступний для користувачів з планшета та персонального комп'ютера, але не зі смартфона;
- 5) доступні інструменти задовольняють лише частину потреб сучасного веб-розробника, але більшість інструментів, все ж таки треба шукати на інших веб-сайтах;
- 6) реєстрація користувачів (незалежних веб-розробників, ентузіастів або студентів вищих навчальних закладів) доступна лише на меншості з представлених веб-сайтів.

Після аналізу наявних веб-сайтів, створення таблиці порівняльної характеристики, головною і основною задачею є врахування всіх функціональних, візуальних, користувацьких можливостей цих веб-проектів, запобігання недоліків, проблем у подальшій реалізації, створення стабільної та багатофункціональної «Інформаційної системи каталогізації інструментів веб-розробника».

1.4 Постановка задачі

Метою роботи за темою «Інформаційна система каталогізації інструментів веб-розробника» - це розробка інформаційної системи, яка буде пропонувати користувачу інструменти веб-розробника, серед яких: інструменти для перевірки IP-адрес, доменів, веб-сайтів, робота з текстом і кодом, кодувальники і декодувальники зашифрованих повідомлень або URL-адрес, мінімізатори і валідатори коду, та яка буде задовільнять наступні потреби:

- 1) бути зручною і простою в використанні, без перенавантаження дизайну;

- 2) мати високий рівень зручності, бути доступною для використання з різних пристроїв, які відповідають потребам веб-розробників;
- 3) мати комплексний набір інструментів для веб-розробників, щоб уникнути потреби у пошуку додаткових інструментів на інших веб-сайтах;
- 4) мати можливість реєстрації користувачів на веб-сайті.

Система повинна дозволяти будь-якому користувачу використовувати наявні інструменти веб-розробника, а також - створити обліковий запис, де користувач зможе користуватись більшою частиною доступних інструментів лише після реєстрації. Для подальшої реалізації поставленої задачі та досягнення цілі в вигляді створення «Інформаційної системи каталогізації інструментів веб-розробника», сформульовані наступні завдання:

- 1) обрати методи та технології для:
 - 1.1) реєстрації та авторизації користувачів у системі;
 - 1.2) реалізації усіх необхідних інструментів веб-розробника;
- 2) спроектувати інформаційну систему та її частини;
- 3) розробити інформаційну систему;
- 4) протестувати систему в цілому.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Інформаційна модель системи каталогізації інструментів

Сучасні веб-сайти є продуктом поєднання технологій, методів і практик, спрямованих на надання користувачам безперебійного, інтерактивного та візуально привабливого досвіду. Вони зазвичай створюються за допомогою мов розмітки HTML і CSS та мов програмування JavaScript, PHP та SQL для управління базами даних.

Створення інформаційної моделі «Інформаційної системи каталогізації інструментів веб-розробника» передбачає розробку та впровадження моделі даних, яка представляє собою різні інструменти, що можуть знадобитися веб-розробнику, а також які будуть об'єднані в єдину інформаційну систему. Інструменти, які ми використовуємо для розробки цієї системи, обрані на основі їх актуальності та ефективності у веб-розробці.

На рис. 2.1 зображено розроблену схему, типової для багатьох веб-сайтів, інформаційної системи з окремими маршрутами виконання операцій з інструментами.

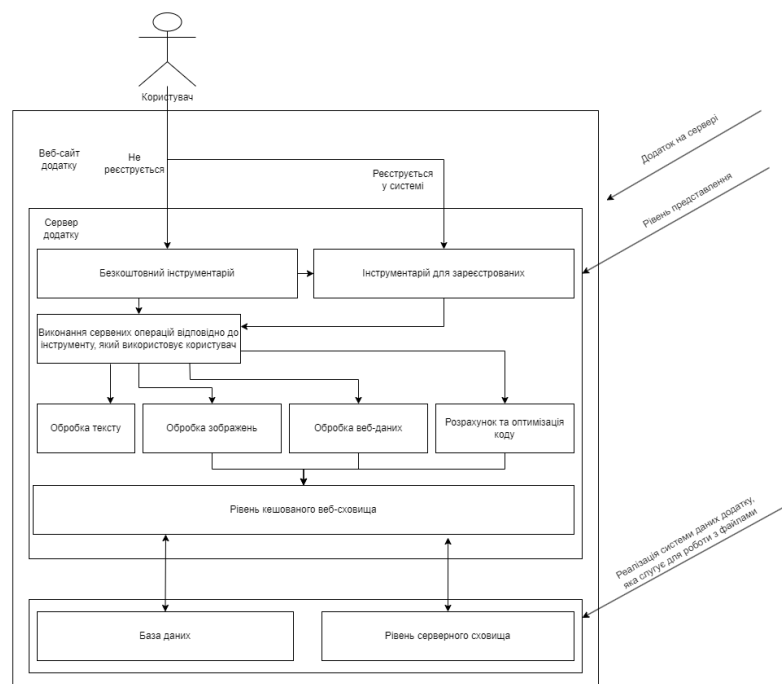


Рисунок 2.1 – Схема інформаційної системи

Цей тип схеми є ідеальним для невеликих додатків, у тому числі і для «Інформаційної системи каталогізації інструментів веб-розробника». Вибір цього варіанту схеми забезпечує баланс функціональності, продуктивності та доступності. Інструменти є загальноприйнятими в галузі, що гарантує сумісність, а також потрібність у будь-який час.

Для структурування та стилізації використовуються HTML та CSS. JavaScript використовується для додавання інтерактивності та покращення користувацького досвіду. PHP, мова на стороні сервера, яка використовується для створення динамічного контенту та обробки запитів користувача (рис. 2.2), а SQL - для взаємодії з базами даних, зберігання та отримання даних.



Рисунок 2.2 – Приклад клієнт-серверних запитів

Використання PHP дозволяє скористатися перевагами широкого спектру бібліотек та інструментів, підвищуючи ефективність та зменшуючи час і зусилля, необхідні для створення інформаційної системи з нуля. SQL, будучи широко використовуваною мовою управління базами даних, гарантує надійну обробку даних і можливості маніпулювання ними швидко та просто.

Головною задачею вищеперечислених інструментів та схем, має на меті забезпечити зручну та організовану платформу для веб-розробників, щоб вони могли знаходити та отримувати доступ до інструментів, які покращують та оптимізують їхній робочий процес. Класифікуючи інструменти за категоріями, надаючи вичерпну інформацію, забезпечуючи пошук і фільтрацію, а також враховуючи відгуки користувачів, система каталогізації має на меті спростити процес пошуку потрібних інструментів для конкретних потреб веб-розробки.

2.2 Прототипування веб-сайту

«Інформаційна система каталогізації інструментів веб-розробника» - це комплексний веб-сайт, призначений для надання веб-розробникам широкого спектру інструментів для підвищення їхньої продуктивності та оптимізації процесу розробки. Веб-сайт складається з трьох основних частин: головної сторінки, сторінок з інструментами та кабінету користувача.

Сторінка «Головна сторінка» - сторінка, яка слугує точкою входу для користувачів, які відвідують сайт. Вона має чистий і зручний дизайн, який легко адаптується до різних пристроїв, включаючи смартфони, планшети та ПК. На сторінці помітно видно поле пошуку, де користувачі можуть вводити певні ключові слова для пошуку потрібних інструментів. Крім того, всі доступні інструменти представлені в організованому вигляді, відсортовані за категоріями. Така категоризація дозволяє розробникам легко орієнтуватися на веб-сайті та знаходити інструменти, що відповідають їхнім конкретним потребам.

Сторінка «Сторінки з інструментами» - сторінка, де обравши інструмент на головній сторінці або здійснивши пошук, користувачі потрапляють на сторінку інструменту. Ця сторінка містить детальну інформацію про обраний інструмент, включаючи його опис, функціональні можливості та інструкції з використання. Поряд з інформацією про інструмент, користувачі також можуть знайти список схожих інструментів, які можуть бути цікавими, а також добірку популярних інструментів, яким надають перевагу інші користувачі. Ці

рекомендації допомагають розробникам знаходити нові інструменти та розширювати свій інструментарій.

Сторінка «Кабінет користувача» - це персоналізований простір, доступний користувачам, які зареєстрували свій обліковий запис на веб-сайті. Після завершення процесу реєстрації, користувачі отримують доступ до «Кабінету користувача», який пропонує додаткові функціональні можливості та варіанти налаштування свого облікового запису. У «Кабінеті користувача», користувачі можуть керувати даними свого облікового запису, наприклад, змінювати свої облікові дані для входу (ім'я користувача або пароль) та оновлювати адресу електронної пошти. Можливість увімкнути двофакторну автентифікацію підвищує безпеку облікового запису користувача, забезпечуючи вищий рівень захисту. Важливо, що реєстрація в «Кабінеті користувача» надає доступ до всіх інструментів, доступних на веб-сайті, що дозволяє користувачам максимально ефективно використовувати його переваги.

2.3 Інструменти, мови програмування, фреймворки

При розробці інформаційної системи було використано навички та знання, які отримані за період навчання у Сумському Державному Університеті, а саме: мова гіпертексту HTML, мова розмітки CSS, мови програмування JavaScript, PHP, а також - SQL.

HTML (HyperText Markup Language) - це мова розмітки гіпертексту, використовувана для створення та оформлення веб-сторінок [8]. HTML визначає структуру документа, розташування тексту, зображень, посилань та інших елементів на веб-сторінці.

CSS (Cascading Style Sheets) - це мова розмітки та опису стилів, яка використовується для задання зовнішнього вигляду веб-сторінок, стилізації та форматування елементів HTML [9]. CSS дозволяє змінювати кольори, шрифти, розташування, розміри та інші атрибути елементів веб-сторінок.

JS (JavaScript) - це скриптова мова програмування, яка використовується для створення динамічних ефектів та інтерактивності на веб-сторінках [10].

JavaScript може взаємодіяти з користувачем, маніпулювати елементами сторінки, обробляти події та виконувати різноманітні дії.

PHP (Hypertext PreProcessor) - це серверна скриптова мова програмування, яка використовується для розробки веб-додатків та динамічних веб-сторінок [11]. PHP може взаємодіяти з базами даних, зчитувати та записувати дані, обробляти форми, генерувати динамічний контент на веб-сторінках.

SQL (Structured Query Language) - це мова запитів до баз даних, яка використовується для створення, модифікації та управління базами даних [12]. SQL дозволяє виконувати запити для отримання, вставки, оновлення та видалення даних з баз даних, а також встановлювати зв'язки між таблицями.

При розробці веб-сайту, було використано програмний додаток Visual Code Studio для роботи з мовами розмітки та мовами програмування.

Visual Studio Code (VS Code) - це безкоштовний редактор коду, розроблений компанією Microsoft [13]. Він є потужним інструментом для редагування та написання програмного коду на різних мовах програмування. Visual Studio Code надає широкі можливості, такі як підсвічування синтаксису, автодоповнення, вбудоване керування версіями, відладка коду та інтеграцію з різними розширеннями, що доповнюють його функціональність. Він має інтуїтивний і зручний інтерфейс користувача, що робить роботу з ним приємною та ефективною. VS Code підтримується на різних операційних системах, включаючи Windows, macOS та Linux, і він став популярним серед розробників програмного забезпечення.

При розробці інтерфейсу інформаційної системи та веб-сайту загалом, було використано фреймворк Bootstrap, який є незамінним інструментом для швидкої розробки проекту будь-якого початківця в веб-програмуванні.

Фреймворк в контексті CSS - це набір готових стилів, компонентів, шаблонів та інструментів, які допомагають створювати веб-інтерфейси швидше та ефективніше [14]. Він надає заздалегідь визначені структури та стилі, які можна використовувати для створення консистентного та професійного вигляду

веб-сайтів. Фреймворки CSS, зазвичай, включають в себе готові класи, стилі кнопок, форм, сітки, навігаційні та інші елементів, що спрощують розробку і полегшують підтримку. Використання фреймворку дозволяє зекономити час, покращити доступність та забезпечити єдність стилів на веб-сайті, що сприяє прискоренню розробки та забезпечує більший контроль над виглядом та поведінкою елементів.

Фреймворк Bootstrap - це відкритий інструментарій для розробки веб-сайтів із застосуванням HTML, CSS і JavaScript [15]. Він надає набір готових компонентів, шаблонів і стилів, які допомагають створювати сучасний, адаптивний та зручний для використання інтерфейс. Bootstrap розроблений з урахуванням принципів адаптивного дизайну, що дозволяє автоматично пристосовувати веб-сторінки до різних розмірів екранів, включаючи мобільні пристрої. Фреймворк Bootstrap використовується багатьма веб-розробниками для прискорення процесу розробки і полегшення стилізації веб-сайтів.

За допомоги HTML, CSS, JavaScript, PHP, SQL та фреймворку Bootstrap був розроблений веб-сайт інформаційної системи. HTML було використано для створення структури веб-сторінок, розташування тексту, зображень та посилань. CSS було використано для задання зовнішнього вигляду сторінок, встановлення кольорів, шрифтів, розмірів та інших атрибутів елементів. JS було використано для створення динамічних ефектів та інтерактивності, взаємодії з користувачем та обробки подій на сторінці. PHP було використано для розробки серверної логіки, зчитування та запису даних з баз даних, обробки форм та генерації динамічного контенту. SQL було використано для створення та керування базою даних, виконання запитів та роботи з даними. Фреймворк Bootstrap надав готові компоненти, шаблони та стилі, що спростили стилізацію та розробку веб-інтерфейсу, забезпечивши швидкий доступ до стандартних елементів та адаптивний дизайн. За допомоги цих інструментів, веб-сайт «Інформаційної системи каталогізації інструментів веб-розробника» був розроблений з

використанням сучасних стандартів та кращих практик розробки, що дозволило забезпечити професійний вигляд, функціональність та ефективну роботу сайту.

За допомоги HTML, CSS, JavaScript, PHP, SQL та фреймворку Bootstrap був розроблений «Кабінет користувача» з авторизацією через cookie-файли. Cookie-файли - це невеликі текстові файли, які зберігаються на комп'ютері користувача і містять інформацію про взаємодію з певним веб-сайтом. Вони використовуються для збереження стану сесії, встановлення налаштувань та збереження інших даних, які можуть використовуватися пізніше.

В інформаційній системі, при авторизації користувача, було використано cookie-файли для збереження ідентифікатора сесії [16] або інших даних, що підтверджують авторизацію. Після успішної авторизації, сервер генерує унікальний ідентифікатор сесії та створює cookie-файл, який містить цей ідентифікатор (рис. 2.3). Після цього, при кожному запиті веб-кабінету, цей cookie-файл надсилається з браузера користувача на сервер, що дозволяє серверу ідентифікувати користувача та зберігати його авторизацію протягом сесії.

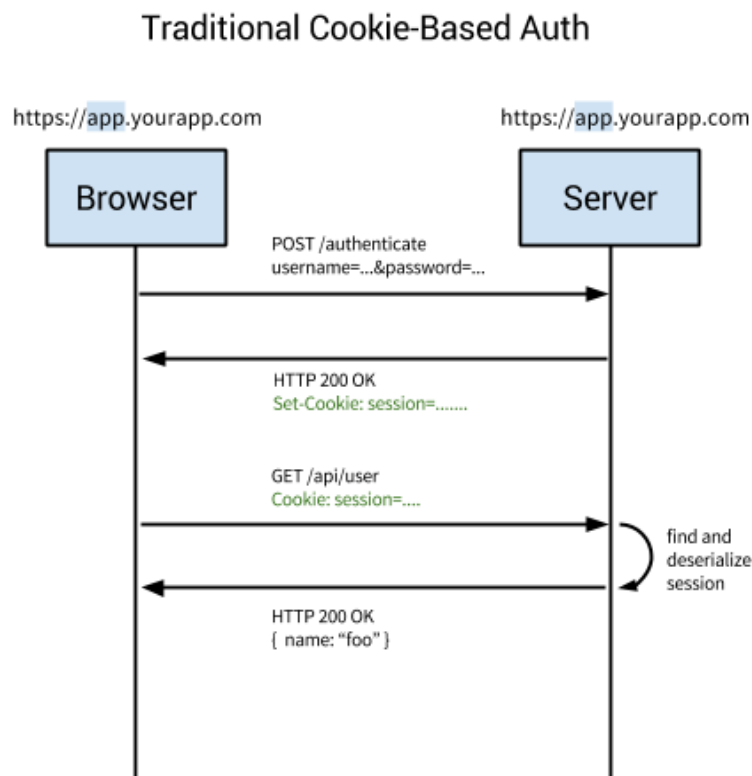


Рисунок 2.3 – Схематичне зображення авторизації на основі cookie-файлів

Авторизація за допомоги cookie-файлів є популярним вибором з кількох причин [17]:

1. **Зручність для користувача:** Після авторизації, інформація про авторизацію зберігається на боці клієнта у вигляді cookie-файлу. Це дозволяє користувачу, не вводити дані авторизації кожного разу, коли він відвідує веб-сайт знову. Користувач може залишатися авторизованим протягом тривалого часу без необхідності повторного введення облікових даних.

2. **Збереження стану сесії:** Cookie-файли дозволяють зберігати ідентифікатори сесій або іншу інформацію, що підтверджує авторизацію на боці клієнта. Це дозволяє серверу ідентифікувати користувача та зберігати його стан і контекст протягом сесії. Наприклад, це дозволяє зберігати обрану мову, налаштування відображення або інші персоналізовані налаштування.

3. **Масштабованість:** Авторизація через cookie-файли дозволяє легко масштабувати систему, особливо при наявності багатьох серверів або фронтенд-балансувальників. Кожен сервер може перевіряти наявність та правильність cookie-файла і виконувати авторизацію незалежно один від одного. Це спрощує горизонтальне масштабування веб-додатків та забезпечує єдність авторизації в усіх частинах системи.

4. **Безпека:** Правильно налаштовані cookie-файли можуть забезпечити додатковий рівень безпеки. Наприклад, їх можна налаштувати з обмеженим терміном життя, шифруванням або використанням токенів, що змінюються з кожним запитом. Це дозволяє ускладнити можливість зламу або підробки авторизаційних даних.

У підсумку, авторизація за допомоги cookie-файлів є потужним інструментом для створення безпечних та зручних веб-додатків. Вона спрощує доступ до системи, зберігає стан сесії та надає можливості для збору та аналізу даних. Проте, необхідно дотримуватися принципів конфіденційності та захисту даних користувачів, щоб забезпечити надійність та довіру до веб-сайтів.

Обрання СУБД (Системи Управління Базами Даних) є важливим кроком при розробці будь-якого проекту. У даному випадку, для інформаційної системи, було обрано СУБД MySQL серед представлених на ринку Oracle DB, Microsoft SQL Server, IBM DB2, MariaDB та Firebird з наступних причин [18]:

1. Простота використання: MySQL відомий своєю простотою та легкістю використання. Він має простий та зрозумілий синтаксис SQL, який робить його доступним для розробників з будь-яким рівнем досвіду.

2. Освіченість та підтримка: MySQL є однією з найпопулярніших СУБД у світі з великою спільнотою користувачів та активною підтримкою. Велика кількість матеріалів, документації та ресурсів доступно для вивчення та розвитку навичок. Крім того, MySQL заснований на відкритому коді, що дозволяє залучати внески та використовувати попередньо перевірений функціонал.

3. Підтримка стандартів: MySQL підтримує багато стандартів SQL і є сумісним з ANSI SQL. Це означає, що ваш код та запити, розроблені для MySQL, будуть працювати на інших СУБД, якщо у вас з'явиться потреба змінити систему в майбутньому [19].

4. Відмінна продуктивність: MySQL відзначається швидкістю та ефективністю роботи з базами даних. Він добре справляється з обробкою великих обсягів даних та запитів, що є важливим для проектів з високим навантаженням.

5. Розширені можливості: MySQL надає розширені функціональні можливості, такі як підтримка реплікації, кластеризації, розподіленого доступу, що дають можливість масштабувати та оптимізувати бази даних залежно від потреб проекту.

Загалом, обрання MySQL є розумним рішенням. MySQL має потужні можливості та підтримку, які допоможуть успішно розробити та впровадити проект з використанням бази даних.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Проектування інформаційної системи

Практична реалізація «Інформаційної системи каталогізації інструментів веб-розробника» повинна включати в себе взаємодію різних компонентів системи, яка допомагає користувачам використовувати наявні інструменти веб-розробника. Система складається з веб-сайту, бекенду, бази даних, і вона дозволяє користувачам зареєструватися, зберігати та отримувати інформацію про різні інструменти веб-розробника (рис. 3.1).

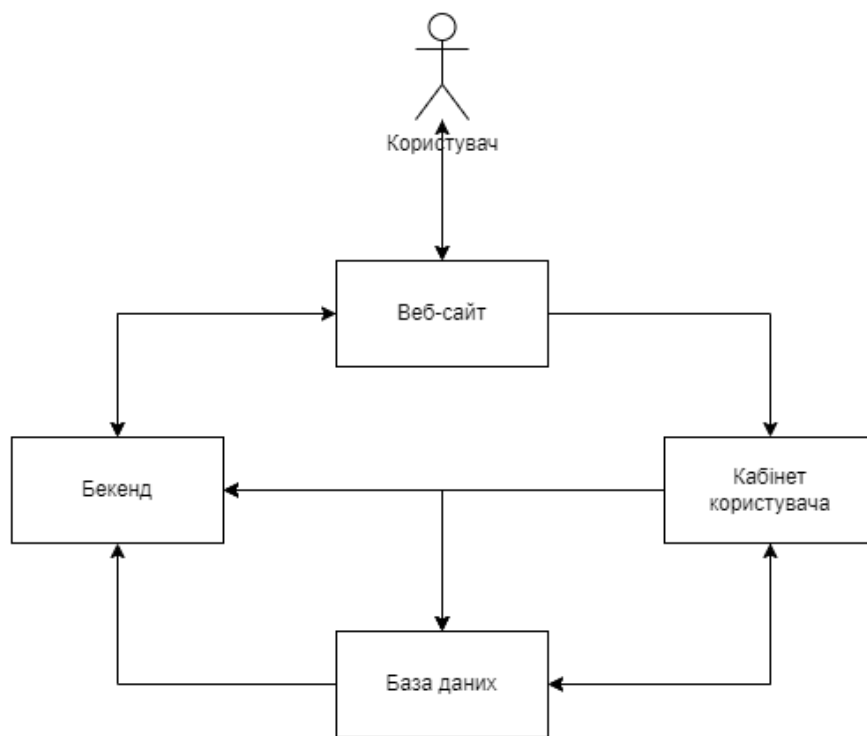


Рисунок 3.1 – Схематичне зображення схеми інформаційної системи

Користувачі взаємодіють з інформаційною системою за допомоги веб-сайту. Веб-сайт містить різні функціональні елементи, включаючи «Кабінет користувача», де вони можуть створити обліковий запис або використовувати існуючий. Користувачі можуть виконувати дії, такі як пошук інструментів, їх використання та створення і редагування облікового запису.

За допомоги веб-сайту, користувач взаємодіє з бекендом системи. Бекенд відповідає за обробку запитів користувача веб-сайту і забезпечує взаємодію з базою даних. Коли користувач виконує будь-яку дію на сторінці інформаційної системи, веб-сайт передає цей запит до бекенду. Бекенд обробляє цей запит, перевіряє стан авторизації користувача (якщо потрібно), взаємодіє з базою даних для збереження або отримання інформації про користувача, і повертає результат.

База даних використовується для зберігання інформації про користувача. Вона містить структуровані дані, описані в таблиці 2.1, такі як ім'я, електронна пошта, пароль, тощо. Бекенд взаємодіє з базою даних для виконання операцій збереження, оновлення, видалення та отримання даних про користувача. Коли веб-сайт або бекенд потребує інформації про користувача, вони звертаються до бази даних через бекенд, який запитує і отримує відповідні дані.

Таблиця 2.1 – Users

Зміст	Опис	Тип	Обмеження	Ключі
user_id	ID користувача	bigint	Not null	PK
email	Пошта користувача	varchar	Not null	
password	Пароль користувача	varchar	Not null	
name	Логін користувача	varchar		
token_code	Токен двох.факт.автор.	varchar		
twofa_secret	Код двох.факт.автор.	varchar		
anti_phishing_code	Антифішинг код	varchar		
one_time_login_code	Одноразовий код	varchar		
pending_email	Очікувана нова пошта	varchar		
email_activation_code	Код активації пошти	varchar		
lost_password_code	Код відновлення паролю	varchar		
status	Стан облікового запису	int	Not null	
language	Мова обл. запису	varchar		
timezone	Таймзона обл. запису	varchar		

ER-діаграма (Entity-Relationship Diagram) є важливою складовою інформаційної системи, бо представляє сутності системи та зв'язки між ними [20]. У випадку «Інформаційної системи каталогізації інструментів веб-розробника», ER-діаграма включає сутності, такі як "Налаштування (settings)", "Користувачі (users)" і "Логи користувачів (users_logs)".

Зв'язки можуть бути визначені, наприклад, як "Користувач використовував веб-сайт з такого пристрою", "Користувач зареєструвався з такої локації" та інше. ER-діаграма (рис. 3.2) допомагає візуалізувати структуру бази даних та взаємозв'язки між сутностями, що сприяє зрозумінню системи та розробці відповідної бази даних.

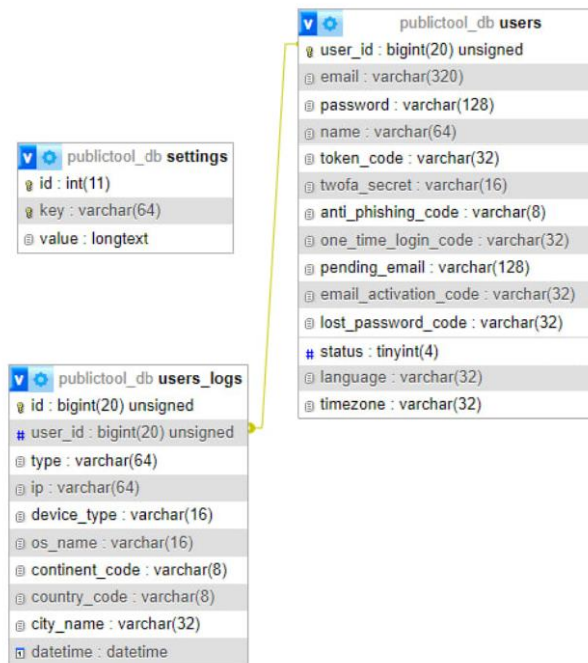


Рисунок 3.2 – ER-діаграма бази даних

Таким чином, ER-діаграма «Інформаційної системи каталогізації інструментів веб-розробника» описує взаємодію між таблицями в базі даних. Вона дозволяє користувачам зареєструватися в «Кабінеті користувача», шукати інструменти веб-розробника та отримувати доступ до цих інструментів через веб-сайт, зв'язаний з бекендом, який взаємодіє з базою даних.

3.2 Програмна реалізація

Після аналізу задач, вимог до інформаційної системи і розгляду доступних аналогів та їх функціональних можливостей, були визначені основні мови програмування для реалізації проекту. HTML було використано для створення структури веб-сторінок та відображення контенту. CSS був використаний для стилізації та оформлення елементів інтерфейсу. JS додав динамічність та інтерактивність до сторінок, обробляючи події та взаємодіючи з користувачем. PHP був використаний для розробки серверної логіки, обробки форм, звернення до бази даних та інших завдань. В якості СУБД була обрана MySQL для збереження та керування даними користувачів.

На початковому етапі розробки, був створений прототип інформаційної системи, який демонстрував основні функціональні можливості, інтерфейс та концепцію веб-сайту. Згодом була розроблена перша версія системи, в якій було реалізовано функції реєстрації, авторизації користувачів і використання базових веб-інструментів. Ця версія була піддана тестуванню та отримала відгуки від перших користувачів для подальшого поліпшення та вдосконалення.

Після процесу врахування отриманих відгуків та вдосконалення, була розроблена готова реалізація інформаційної системи. Ця версія системи включає повний функціонал, розроблений на основі вимог до інформаційної системи. Користувачі мали можливість створювати облікові записи, авторизуватися в системі та використовувати усі наявні інструменти веб-розробника. Була забезпечена безпека даних, обмеження доступу та правильна обробка помилок.

Було розроблено головну сторінку (рис. 3.3), де користувач має можливість використовувати функцію пошуку, використовувати доступні різноманітні інструменти, які дозволяють користувачу оптимізувати процес веб-розробки та виконувати різні операції з наявним на веб-сайті функціоналом.

Головна сторінка інформаційної системи створена з урахуванням зручності і простоти використання. Вона надає швидкий доступ до пошуку та набору інструментів, що дозволяє користувачам ефективно працювати з системою та

займатися розробкою веб-проектів. Завдяки головній сторінці, користувачі можуть швидко знайти необхідний інструмент без зайвих зусиль та затримок.

Пошукова функція надає зручний спосіб знаходження необхідного інструменту веб-розробника в інформаційній системі. Користувач може вводити ключові слова і система проводить пошук задля знаходження відповідних інструментів (рис. 3.4). Це дозволяє швидко та ефективно знаходити потрібну інформацію без необхідності вручну переглядати всі сторінки веб-сайту.

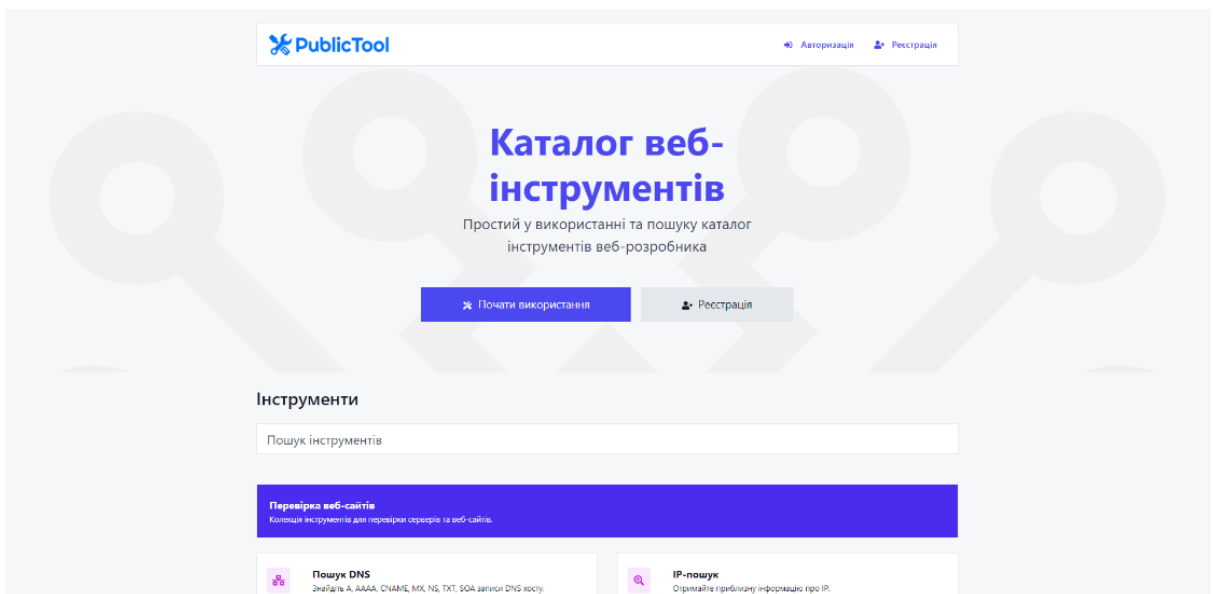


Рисунок 3.3 – Головна сторінка інформаційної системи

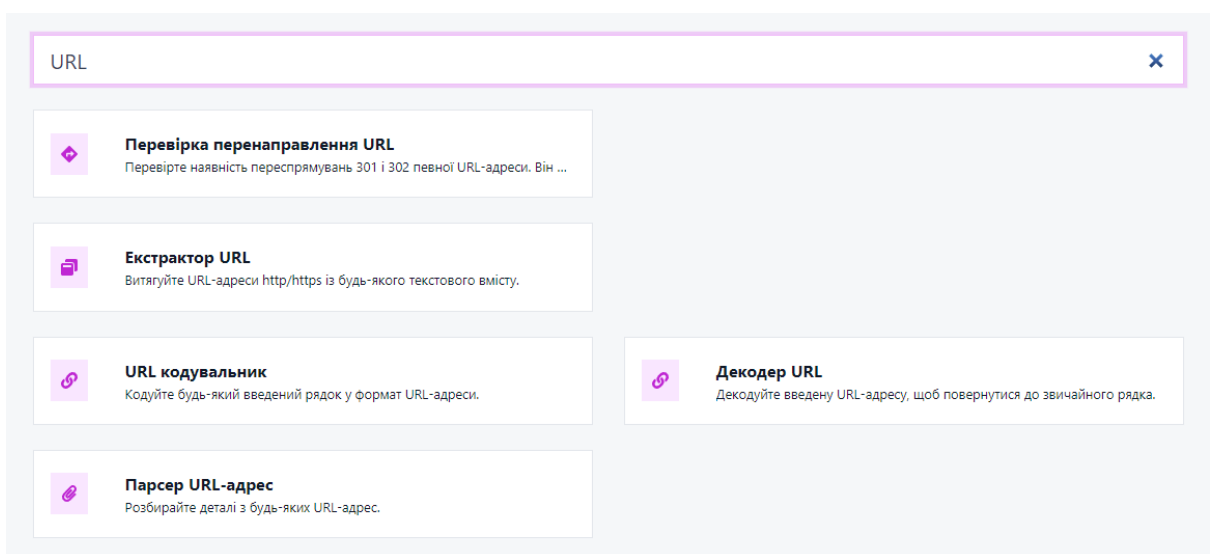


Рисунок 3.4 – Пошук серед інструментів інформаційної системи

Крім того, на головній сторінці доступні наявні інструменти в інформаційній системі (рис. 3.5-3.8). Ці інструменти дозволяють користувачу виконувати різні завдання в житті веб-розробника, наприклад: оптимізувати код, працювати з текстом, перевіряти веб-сайти, домени, тощо.

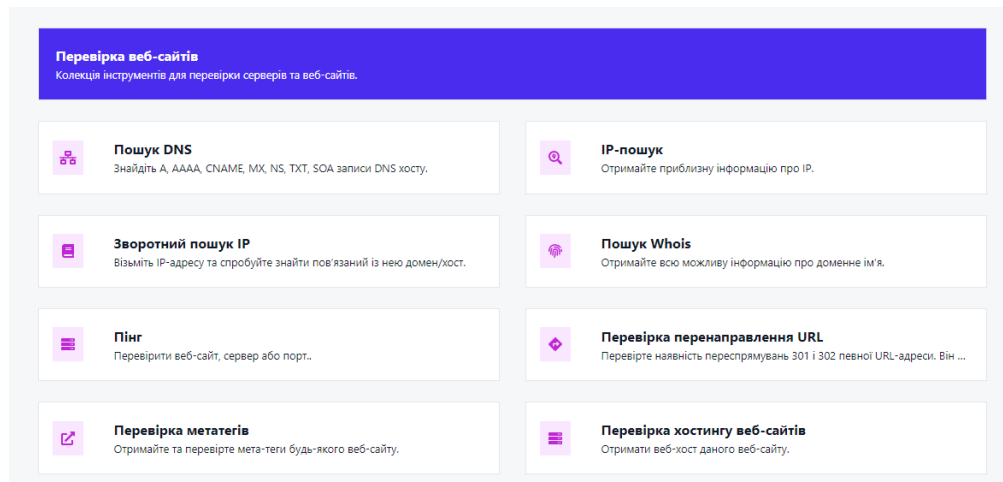


Рисунок 3.5 – Інструменти для перевірки веб-сайтів

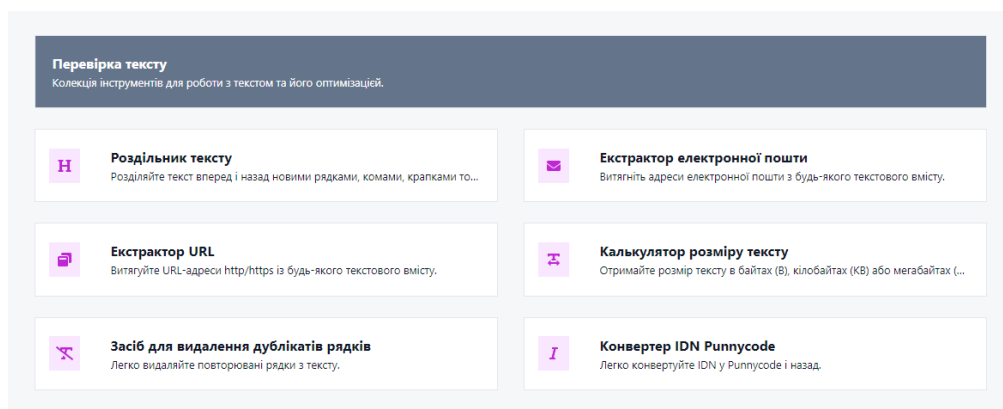


Рисунок 3.6 – Інструменти для перевірки тексту

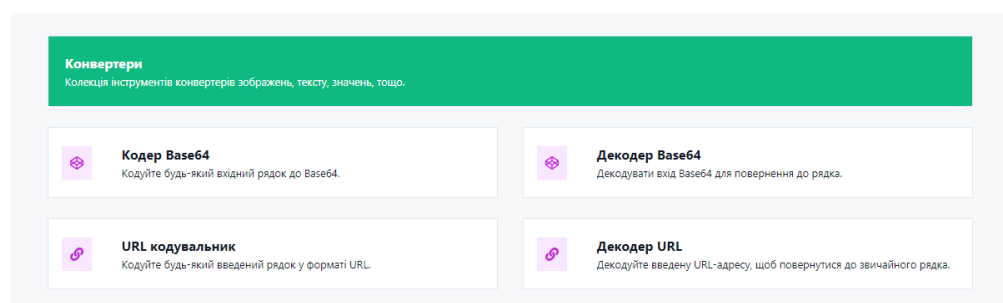


Рисунок 3.7 – Інструменти для конвертації значень

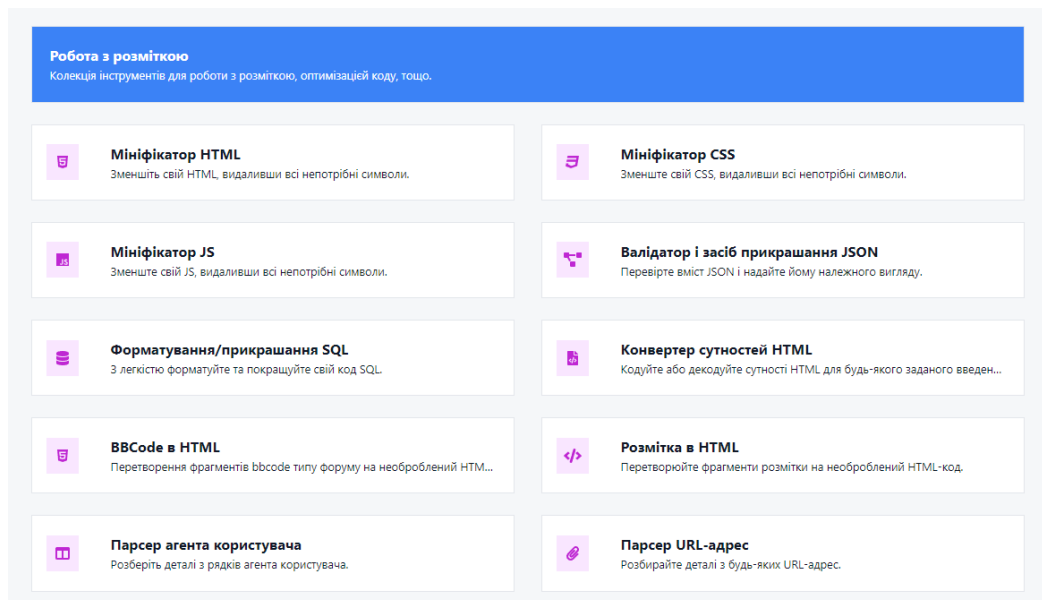


Рисунок 3.8 – Інструменти для роботи з розміткою

На головній сторінці інформаційної системи, розташований навібар, який надає зручний доступ до сторінок авторизації та реєстрації облікового запису. Навбар є навігаційним елементом, який містить посилання та вкладки, що спрямовують користувача на відповідні сторінки для виконання цих дій.

Користувач може клікнути на відповідне посилання, щоб перейти на сторінку авторизації або реєстрації (рис. 3.9).

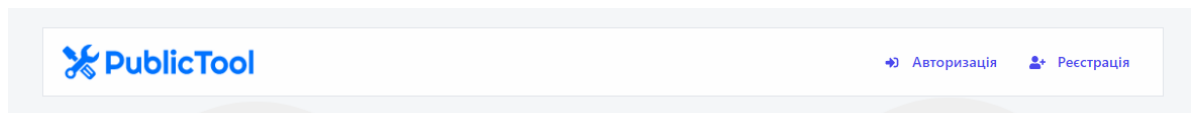


Рисунок 3.9 – Навбар проекту

Сторінка "Авторизація" (рис. 3.10) є важливою частиною інформаційної системи, яка надає користувачам можливість увійти в свій обліковий запис. На цій сторінці розташовані різні елементи, що забезпечують зручний та безпечний процес авторизації.

Користувач повинен ввести свою зареєстровану електронну пошту в поле "Пошта" та пароль в поле "Пароль". Ці поля дозволяють ідентифікувати користувача та перевірити його достовірність для надання доступу до системи.

Рисунок 3.10 – Сторінка «Авторизація»

Капча - це додатковий захисний елемент, що має за мету запобігти автоматизованому доступу до системи шляхом перевірки, що користувач є людиною, а не роботом [21].

На сторінці "Авторизація" присутня кнопка "Запам'ятати мене". Ця опція дає можливість користувачеві зберегти свої дані в системі, щоб його авторизаційні дані заповнювалися автоматично при наступному вході до системи [22].

Також на сторінці присутнє посилання "Відновлення паролю" (рис. 3.11), яке веде до сторінки, де користувач може відновити свій забутий пароль.

Рисунок 3.11 – Сторінка «Відновлення паролю»

Сторінка "Реєстрація" є важливим етапом в інформаційній системі, де користувач може створити свій обліковий запис (рис. 3.12). На цій сторінці розміщені різні елементи, що забезпечують зручний та безпечний процес реєстрації.

Користувач повинен ввести свою поштову адресу в поле "Пошта" та обрати надійний пароль для свого облікового запису в полі "Пароль". Далі розташоване поле капчі, де користувач повинен ввести код з зображення, яке вимагається для підтвердження реєстрації.

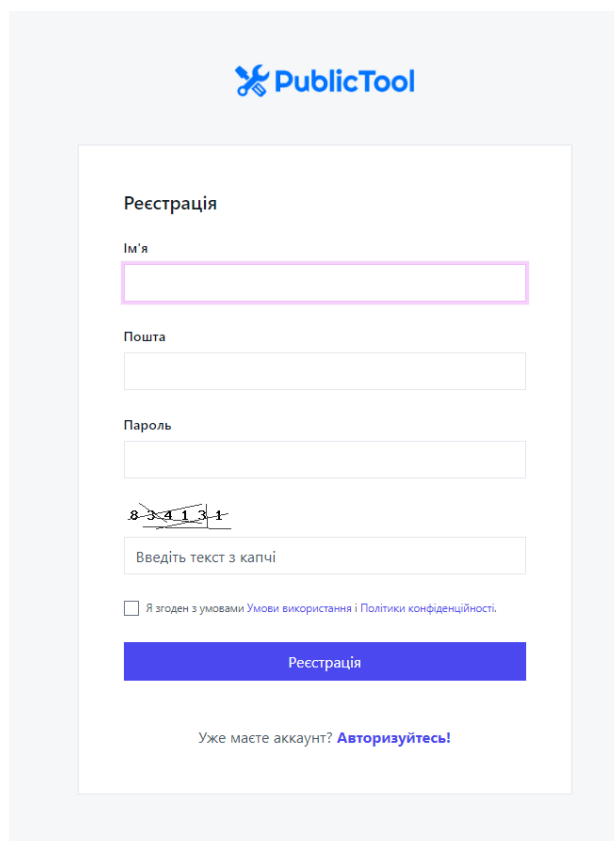


Рисунок 3.12 – Сторінка «Реєстрація»

Останнім елементом сторінки "Реєстрація" є кнопка "Реєстрація". Користувач натискає на цю кнопку після введення необхідних даних, щоб відправити запит на створення облікового запису. Система перевіряє введені дані, переконується у їх правильності та створює новий обліковий запис для користувача і перенаправляє на сторінку «Кабінет користувача».

Сторінка «Кабінет користувача» (рис. 3.13) є центральною точкою управління особистим обліковим записом користувача в інформаційній системі. На цій сторінці доступні різні налаштування, функції для підвищення безпеки та контролю над обліковим записом. Сторінка «Кабінет користувача» містить можливість зміни імені облікового запису, електронної пошти та часового поясу.

Рисунок 3.13 – Сторінка «Кабінет користувача»

Користувач може змінити дані відповідно до своїх потреб та вимог, а також - може встановити антифiшинг код для додаткового захисту свого облікового запису на випадок фiшингових розсилок. Цей код використовується для підтвердження аутентичності розсилки та запобігання фiшинговим атакам.

Сторінка «Кабінет користувача» надає користувачу можливість увімкнути двофакторну авторизацію для свого облікового запису. Цей інструмент додає додатковий рівень безпеки, вимагаючи введення додаткового коду та використання мобільного додатку для підтвердження доступу.

Рисунок 3.14 – Функція «Двофакторна авторизація»

На сторінці «Кабінет користувача», користувач може змінити свій пароль для забезпечення безпеки свого облікового запису. Ця функція дозволяє змінювати пароль на регулярній основі або в разі потреби.

Оновити пароль ⓘ

Старий пароль

Новий пароль

Повторення нового паролю

Оновити

Рисунок 3.15 – Функція «Оновити пароль»

Сторінка «Кабінет користувача» також надає можливість переглядати «Логи», де користувач може переглянути список дій, виконаних з його обліковим записом (рис. 3.16), наприклад, вхід до системи, зміна налаштувань тощо. Це допомагає користувачу відслідковувати активність та забезпечує контроль над обліковим записом.

Обліковий запис | Логи | Видалити аккаунт?

Журнали облікових записів ⓘ

Тип	IP	Подробиці
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 15 хвилин тому
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 33 хвилини тому
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 4 дні тому
login.wrong_password	37...	📄 ☰ ⌂ 🇺🇦 🏠 4 дні тому
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 4 дні тому
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 4 дні тому
login.default.success	37...	📄 ☰ ⌂ 🇺🇦 🏠 4 дні тому

Показано 1 - 7 із 7 результатів.

Рисунок 3.16 – Сторінка «Логи»

На сторінці «Кабінет користувача», користувач має можливість видалити свій обліковий запис, якщо це необхідно (рис. 3.17). Це є важливим аспектом, який надає користувачеві повний контроль над своїми особистими даними та можливістю вибору щодо використання системи.

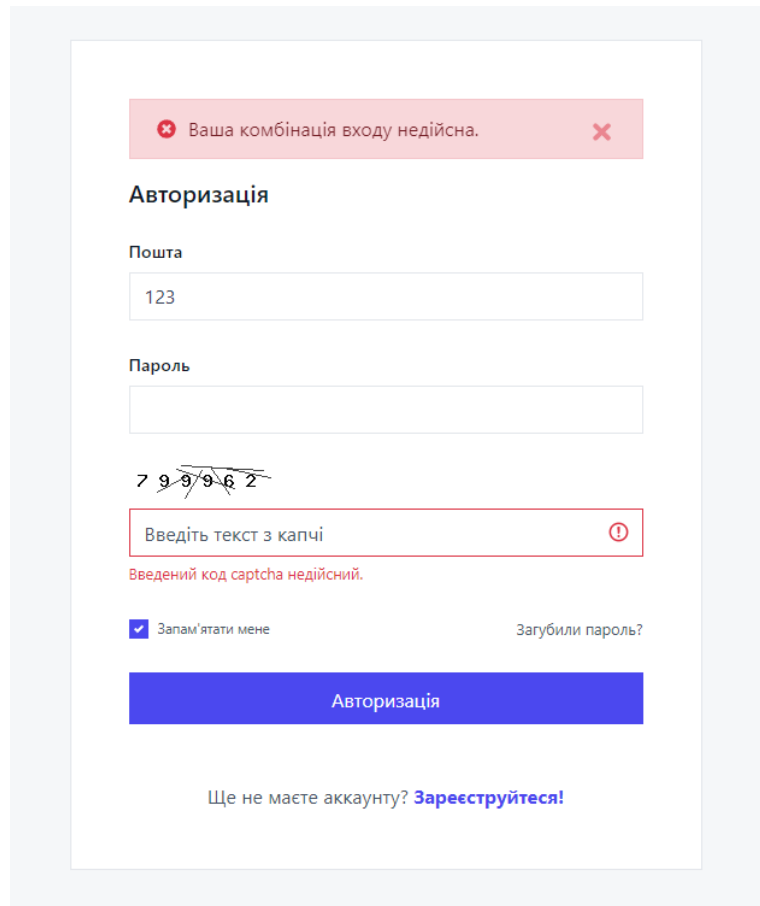
Рисунок 3.17 – Сторінка «Видалення аккаунту»

3.3 Тестування інструментів інформаційної системи

Перевірка інструментів «Інформаційної системи каталогізації інструментів веб-розробника» була проведена з метою впевнитися в її працездатності, безпеці та коректності роботи. Під час тестування були використані різні тестові сценарії використання інструментів, які охоплювали всі важливі аспекти інформаційної системи.

Один з основних тестових сценаріїв полягав у реєстрації нового користувача. Було перевірено, чи відбувається успішна реєстрація, чи правильно зберігаються дані користувача та чи працюють механізми перевірки правильності введених даних (рис. 3.18). Також було виконано тестування на випадок невірного введення даних, щоб переконатися, що система правильно реагує на помилки та надає відповідні повідомлення користувачу.

Після успішної реєстрації, було проведено тестування авторизації. Перевірено, чи працює процес авторизації, чи перевіряються введені дані користувача та чи надається доступ до особистого кабінету тільки авторизованим користувачам. Також було перевірено функціональність кнопки "Запам'ятати мене" та застосування двофакторної авторизації (рис. 3.19).



✖ Ваша комбінація входу недейсна. ✖

Авторизація

Пошта

Пароль

7 9 9 9 6 2

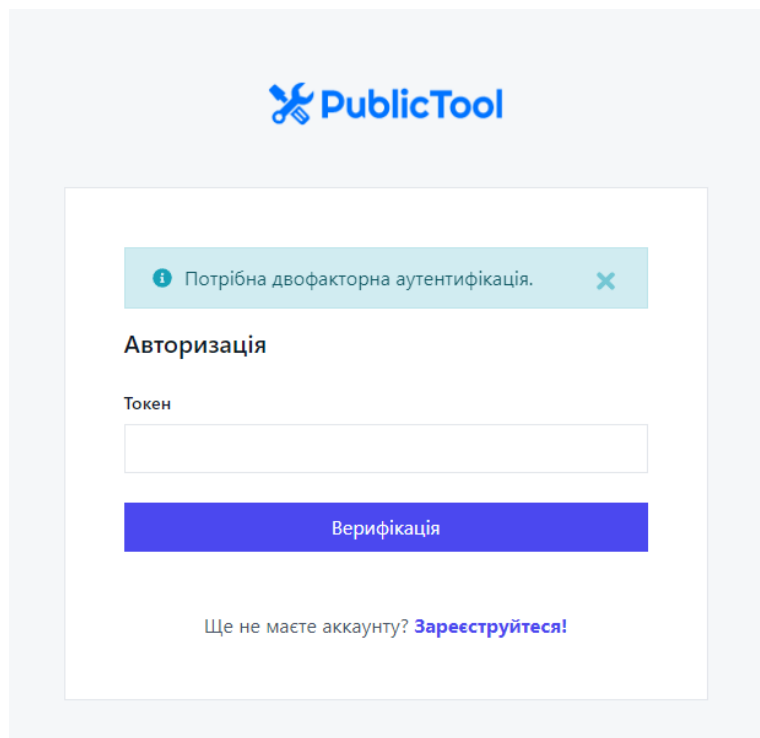
 ⓘ
Введений код сартча недейсний.


Запам'ятати мене [Загубили пароль?](#)

Авторизація

Ще не маєте акаунту? [Зареєструйтеся!](#)

Рисунок 3.18 – Приклад помилки при авторизації



 PublicTool

ℹ Потрібна двофакторна аутентифікація. ✖

Авторизація

Токен

Верифікація

Ще не маєте акаунту? [Зареєструйтеся!](#)

Рисунок 3.19 – Приклад двофакторної авторизації

Окрема увага була приділена тестуванню функціоналу «Інформаційної системи каталогізації інструментів веб-розробника». Перевірено функціонал інструментів для роботи з веб-сайтами, текстом, конвертації, розмітки.

Пошук DNS (рис. 3.20): Інструмент, що відображає записи DNS-серверів для конкретного домену, допомагаючи встановити зв'язок між доменом і сервером.

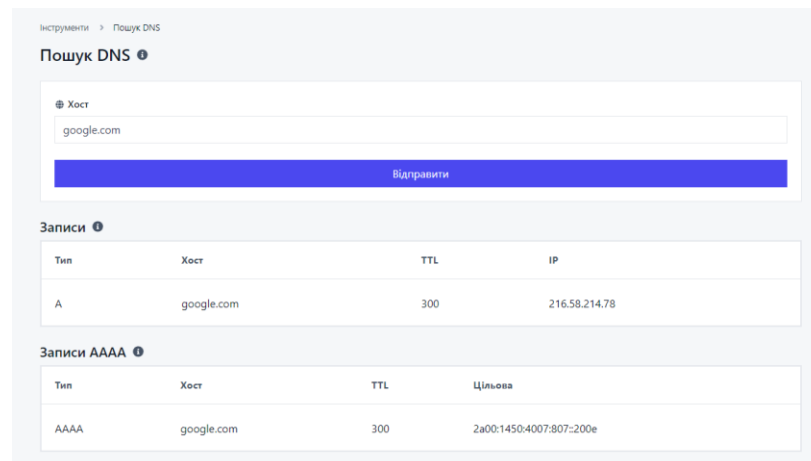


Рисунок 3.20 – Тестування інструменту «Пошук DNS»

IP-пошук (рис. 3.21): Інструмент, який надає інформацію про задану IP-адресу, включаючи географічне розташування, постачальника послуг Інтернету та інші технічні деталі.

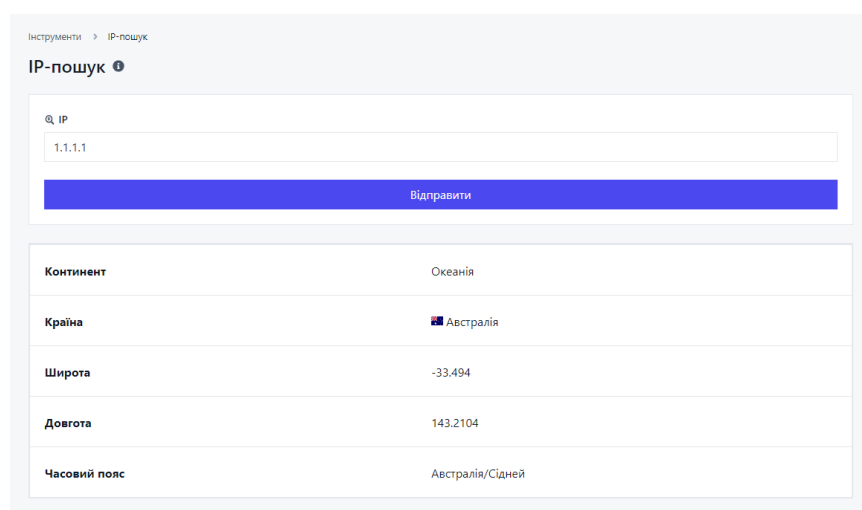


Рисунок 3.21 – Тестування інструменту «IP-пошук»

Зворотній пошук IP (рис. 3.22): Інструмент, що перетворює доменні імена в IP-адреси та навпаки, допомагаючи встановити зв'язок між доменом і сервером.

Інструменти > Зворотний пошук IP

Зворотний пошук IP

🔍 IP

1.1.1.1

Відправити

Результат

one.one.one.one

Рисунок 3.22 – Тестування інструменту «Зворотній пошук IP»

Пошук Whois (рис. 3.23): Інструмент, що надає публічну інформацію про реєстратора доменного імені, технічні деталі та дату реєстрації.

Пошук Whois

🌐 Доменне ім'я

google.com

Відправити

Реєстратор	MarkMonitor, Inc.
Видано на	15 вересня 1997 (1997-09-15 07:00:00)
Оновлено	9 вересня 2019 (2019-09-09 15:39:04)
Термін дії закінчується	13 вересня 2028 (2028-09-13 07:00:00)
Сервери імен	ns1.google.com ns2.google.com ns3.google.com ns4.google.com

Рисунок 3.23 – Тестування інструменту «Пошук Whois»

Пінг (рис. 3.24): Інструмент, який вимірює час, необхідний для передачі пакету даних від вашого пристрою до вказаної IP-адреси або доменного імені, допомагаючи перевірити доступність та швидкість з'єднання.

Статус	✔ Вгору!
Час реакції	162,010 мс
Код статусу відповіді	200

Рисунок 3.24 – Тестування інструменту «Пінг»

Перевірка перенаправлення URL (рис. 3.25): Інструмент, який перенаправляє веб-адресу (URL) з одного ресурсу на інший. Це корисно для зміни URL, збереження SEO-рейтингу та оновлення посилань без зміни видимої адреси для користувачів.

#	URL	Код стану
1	http://google.com	301
2	http://www.google.com/	200

Рисунок 3.25 – Тестування інструменту «Перевірка перенаправлення URL»

Перевірка мета-тегів (рис. 3.26): Інструмент, який аналізує HTML-сторінку та витягує мета-теги, такі як заголовок, опис, ключові слова, що допомагає встановити правильність їх використання для пошукової оптимізації та відображення в результатах пошуку.

Перевірка мета-тегів

URL
https://www.facebook.com

Відправити

referrer	default
og:site_name	Facebook
og:url	https://www.facebook.com/
og:image	https://www.facebook.com/images/fb_icon_325x325.png
og:locale	en_US
description	Log into Facebook to start sharing and connecting with your friends, family, and people you know.

Рисунок 3.26 – Тестування інструменту «Перевірка мета-тегів»

Перевірка хостингу веб-сайтів (рис. 3.27): Інструмент, який допомагає перевірити, на якому хостинг-провайдері (сервері) розміщений веб-сайт, включаючи інформацію про місцезнаходження, доступність та технічні деталі, що допомагає оцінити інформацію про хостинг веб-сайту.

Перевірка хостингу веб-сайтів

Хост
google.com

Відправити

ISP	Google LLC
організація	Google LLC
Країна	Сполучені Штати
Місто	Рівертон
Широта	40.4953
Довгота	-111.9439
Часовий пояс	Америка/Денвер

Рисунок 3.27 – Тестування інструменту «Перевірка хостингу веб-сайтів»

Роздільник тексту (рис. 3.28): Інструмент, який допомагає розділити текстовий рядок на окремі частини за певними критеріями, такими як символ, пробіл, кома або інші роздільники, для подальшої обробки або аналізу даних.

Інструменти > Роздільник тексту

Роздільник тексту

Текст

Лорем;
Імпсум;

Розділяти текст за символом

;

Замінити на

-

Відправити

Результат

Лорем-
Імпсум-

Рисунок 3.28 – Тестування інструменту «Роздільник тексту»

Екстрактор електронної пошти (рис. 3.29): Інструмент, який виявляє та виділяє електронні адреси з тексту або повідомлення, дозволяючи швидко збирати адреси електронної пошти для подальшого використання.

Інструменти > Екстрактор електронної пошти

Екстрактор електронної пошти

Текст

Lorem Ipsum google@google.com

Відправити

Усього

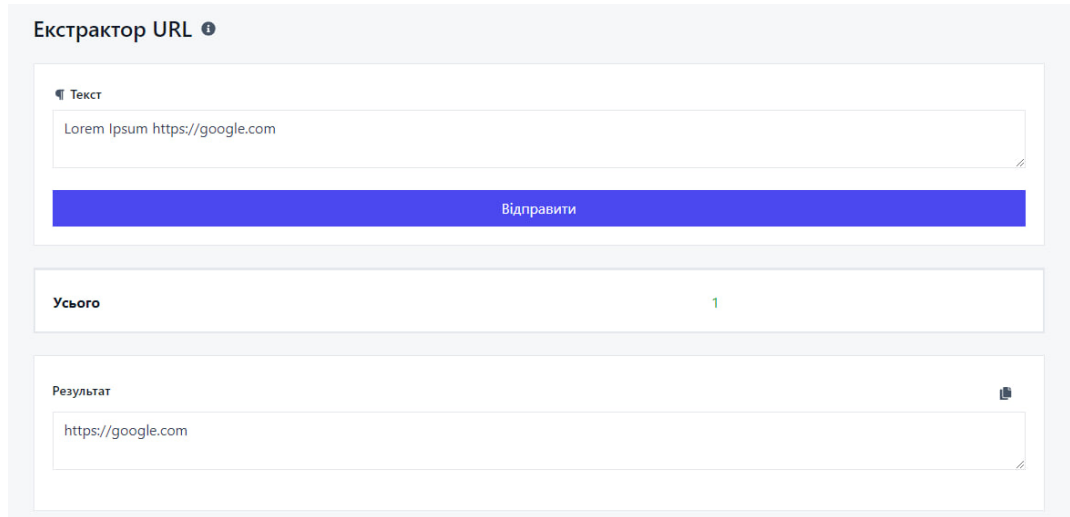
1

Результат

google@google.com

Рисунок 3.29 – Тестування інструменту «Екстрактор електронної пошти»

Екстрактор URL (рис. 3.30): Інструмент, який виявляє та виділяє веб-адреси (URL) з тексту або повідомлення, допомагаючи збирати URL-адреси для подальшого використання або аналізу.



The screenshot shows the 'Екстрактор URL' (URL Extractor) interface. It features a text input field containing 'Lorem Ipsum https://google.com'. Below the input is a blue 'Відправити' (Send) button. A summary bar below the button shows 'Усього' (Total) as '1'. The 'Результат' (Result) section displays the extracted URL 'https://google.com'.

Рисунок 3.30 – Тестування інструменту «Екстрактор URL»

Калькулятор розміру тексту (рис. 3.31): Інструмент, що обчислює кількість символів в байтах в заданому тексті, допомагаючи оцінити його обсяг або виконати підрахунки для редагування, лімітів або аналітики текстових даних.



The screenshot shows the 'Калькулятор розміру тексту' (Text Size Calculator) interface. It features a text input field containing the number '1234567890'. Below the input is a summary bar showing 'Розмір' (Size) as '10 Б' (10 bytes).

Рисунок 3.31 – Тестування інструменту «Калькулятор розміру тексту»

Засіб для видалення дублікатів рядків (рис. 3.32): Інструмент, який виявляє та видаляє повторні рядки з набору даних, допомагаючи зберегти лише унікальні записи та спростити аналіз даних.

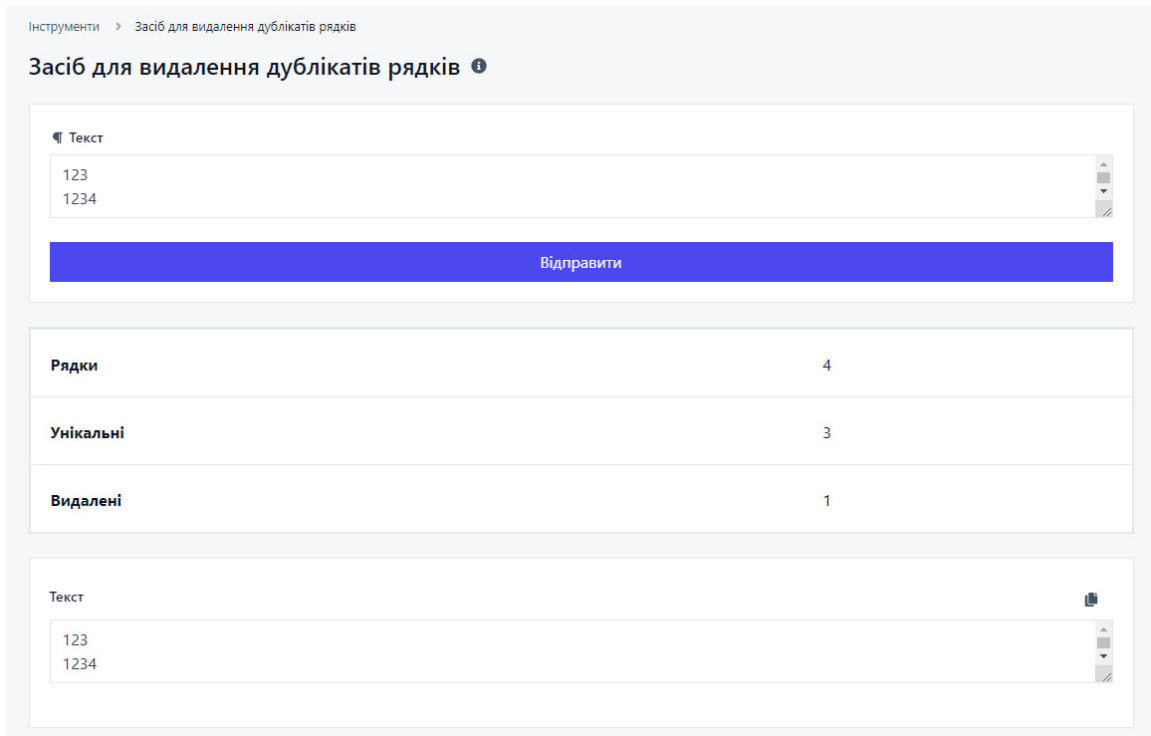


Рисунок 3.32 – Тестування інструменту «Засіб для видалення дублікатів рядків»

Конвертер IDN Punnycode (рис. 3.33): Інструмент, який перетворює міжнародні доменні імена (IDN) у пунктуаційний код (Punnycode) і навпаки, дозволяючи зручно перетворювати IDN-імена для використання в адресних рядках браузера або інших ситуаціях, де потрібне його кодування/декодування.

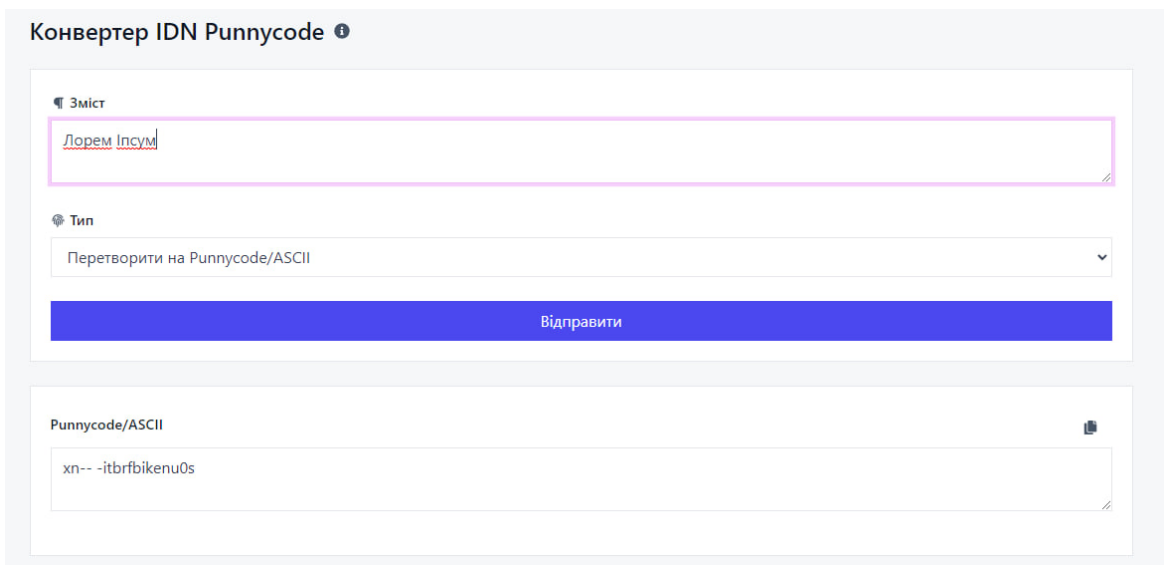
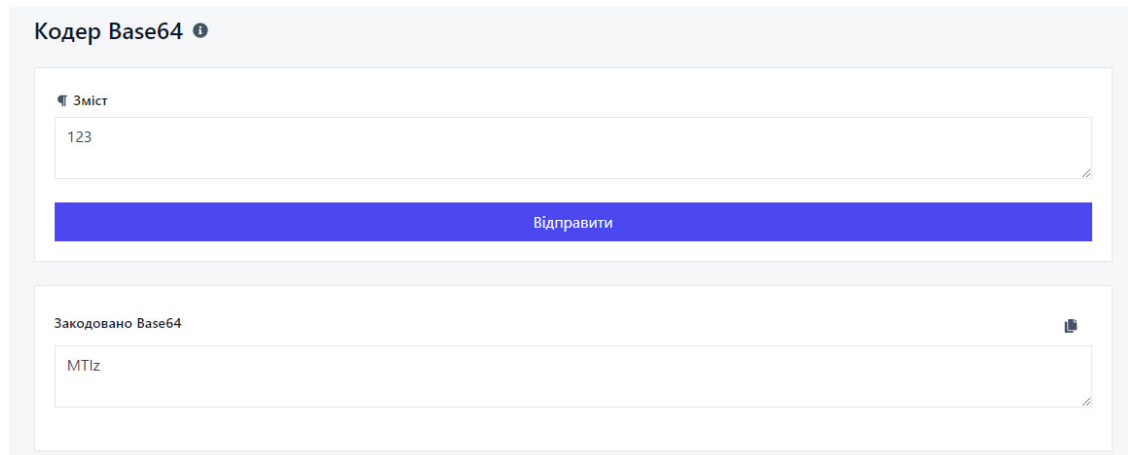


Рисунок 3.33 – Тестування інструменту «Конвертер IDN Punnycode»

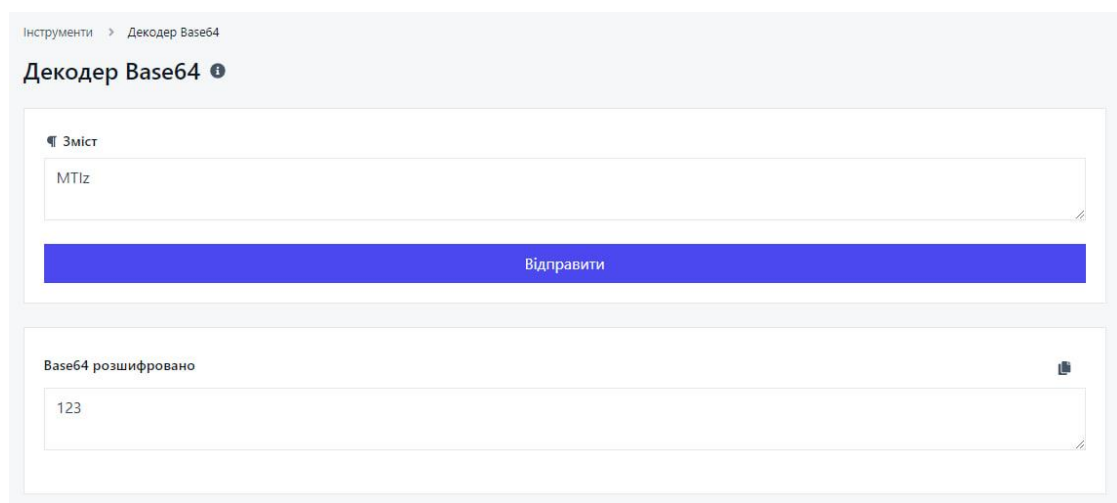
Кодер Base64 (рис. 3.34): Інструмент, який перетворює дані (текст, зображення, файли) в рядок символів, використовуючи кодування Base64. Це дозволяє представити дані у вигляді тексту, який можна передавати без втрати чи пошкодження інформації.



The screenshot shows the 'Кодер Base64' (Base64 Encoder) interface. It features a title bar with the text 'Кодер Base64' and an information icon. Below the title bar is a section labeled 'Зміст' (Content) with a text input field containing '123'. A blue button labeled 'Відправити' (Send) is positioned below the input field. The bottom section is labeled 'Закодовано Base64' (Encoded Base64) and contains a text output field displaying 'MTIz'.

Рисунок 3.34 – Тестування інструменту «Кодер Base64»

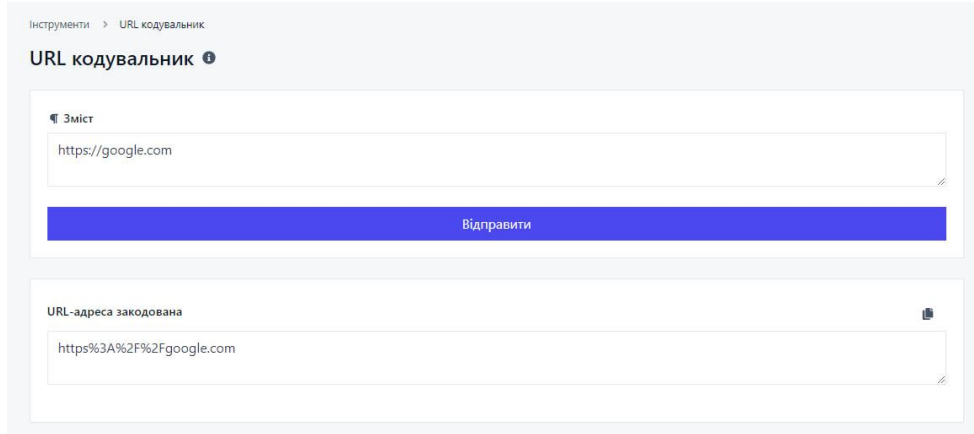
Декодер Base64 (рис. 3.35): Інструмент, який розшифровує рядок символів, закодований у форматі Base64, і відновлює його в початковий вигляд даних (текст, зображення, файли). Це дозволяє отримати початкові дані з закодованого рядка.



The screenshot shows the 'Декодер Base64' (Base64 Decoder) interface. The breadcrumb path at the top reads 'Інструменти > Декодер Base64'. The title bar contains 'Декодер Base64' and an information icon. The 'Зміст' (Content) section has a text input field with 'MTIz' and a blue 'Відправити' (Send) button. The 'Base64 розшифровано' (Base64 Decoded) section shows the output field containing '123'.

Рисунок 3.35 – Тестування інструменту «Декодер Base64»

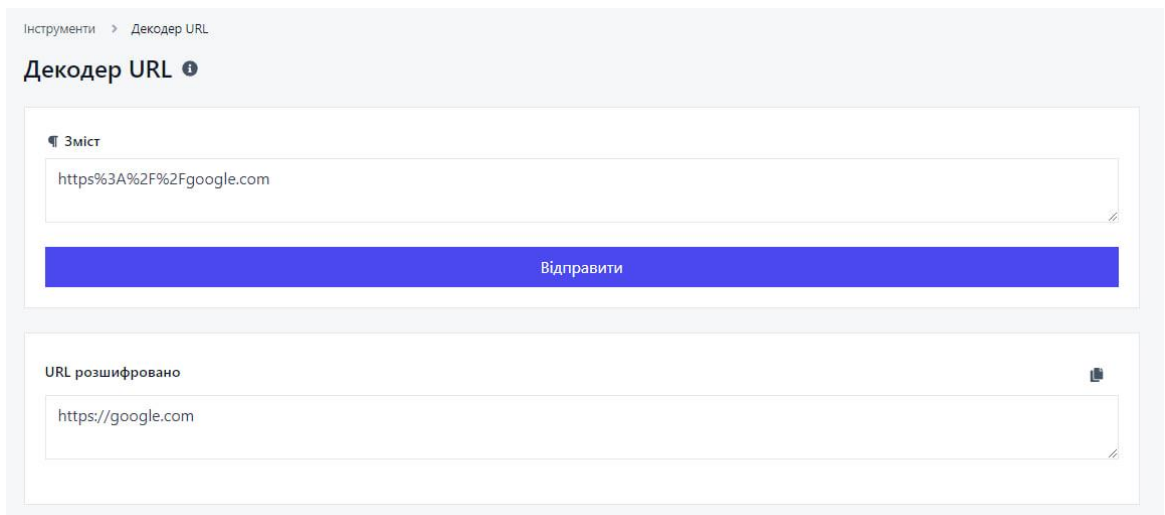
URL кодувальник (рис. 3.36): Інструмент, який перетворює спеціальні та недопустимі символи в URL-адресах в їх кодовану форму, щоб забезпечити коректність та безпеку передачі даних через Інтернет.



The screenshot shows the 'URL кодувальник' (URL Encoder) tool interface. At the top, it says 'Інструменти > URL кодувальник'. Below that is the title 'URL кодувальник' with an information icon. There are two main input sections. The first section is labeled 'Зміст' (Content) and contains a text box with the URL 'https://google.com'. Below this text box is a blue button labeled 'Відправити' (Send). The second section is labeled 'URL-адреса закодована' (URL address encoded) and contains a text box with the encoded URL 'https%3A%2F%2Fgoogle.com'. There is a copy icon to the right of this text box.

Рисунок 3.36 – Тестування інструменту «URL кодувальник»

Декодер URL (рис. 3.37): Інструмент, який розшифровує закодовані символи в URL-адресі і повертає їх до початкового стану. Це дозволяє отримати оригінальну форму URL-адреси після кодування.



The screenshot shows the 'Декодер URL' (URL Decoder) tool interface. At the top, it says 'Інструменти > Декодер URL'. Below that is the title 'Декодер URL' with an information icon. There are two main input sections. The first section is labeled 'Зміст' (Content) and contains a text box with the encoded URL 'https%3A%2F%2Fgoogle.com'. Below this text box is a blue button labeled 'Відправити' (Send). The second section is labeled 'URL розшифровано' (URL decoded) and contains a text box with the decoded URL 'https://google.com'. There is a copy icon to the right of this text box.

Рисунок 3.37 – Тестування інструменту «Декодер URL»

Мініфікатор HTML (рис. 3.38): Інструмент, який зменшує розмір HTML-коду шляхом видалення зайвих пробілів, відступів та коментарів, що полегшує передачу та завантаження сторінок, але зберігає функціональність та структуру HTML-коду.

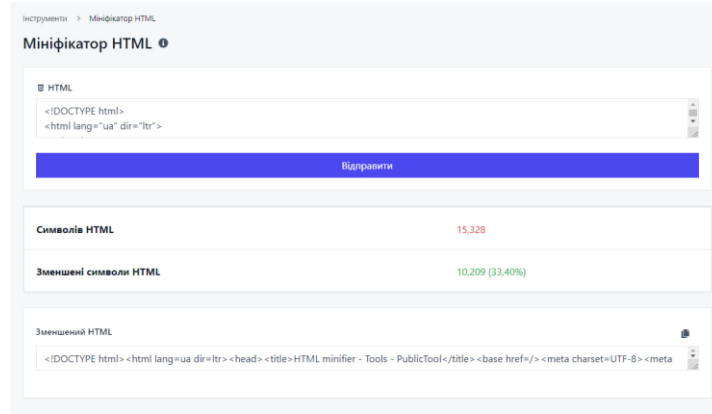


Рисунок 3.38 – Тестування інструменту «Мініфікатор HTML»

Мініфікатор CSS (рис. 3.39): Інструмент, який зменшує розмір CSS-коду шляхом видалення зайвих пробілів, коментарів, перенесень рядків та інших непотрібних символів, зберігаючи при цьому функціональність та стиль CSS-стилей. Це допомагає покращити швидкість завантаження сторінок та оптимізувати розмір файлів CSS.

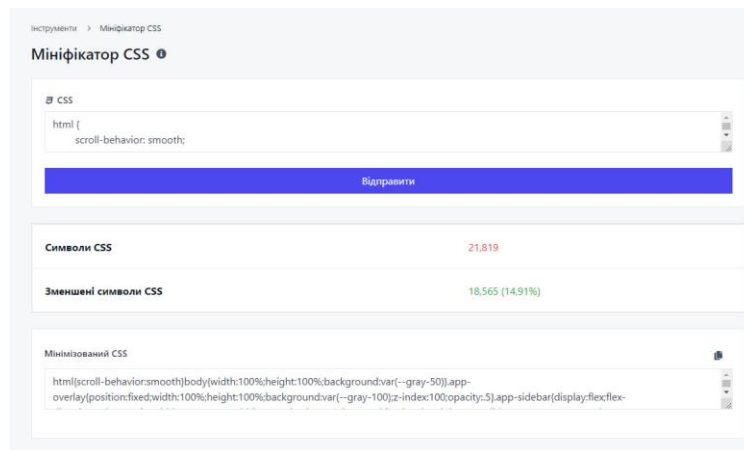


Рисунок 3.39 – Тестування інструменту «Мініфікатор CSS»

Мініфікатор JS (рис. 3.40): Інструмент, який зменшує розмір JavaScript-коду шляхом видалення зайвих пробілів, коментарів, перенесень рядків та інших непотрібних символів, зберігаючи при цьому функціональність та логіку коду. Це сприяє покращенню швидкості завантаження сторінок та зменшенню розміру файлів JavaScript.

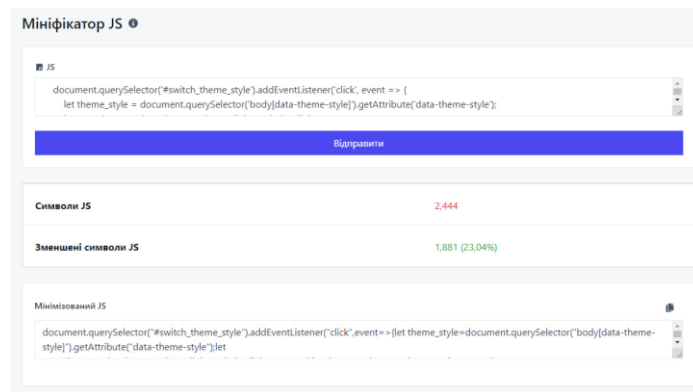


Рисунок 3.40 – Тестування інструменту «Мініфікатор JS»

Валідатор і засіб прикрашання JSON (рис. 3.41): Інструмент, який перевіряє правильність синтаксису JSON-структур та надає зрозуміле форматування для полегшення читабельності. Це допомагає виявляти та виправляти помилки у JSON-даних та поліпшує організацію коду.

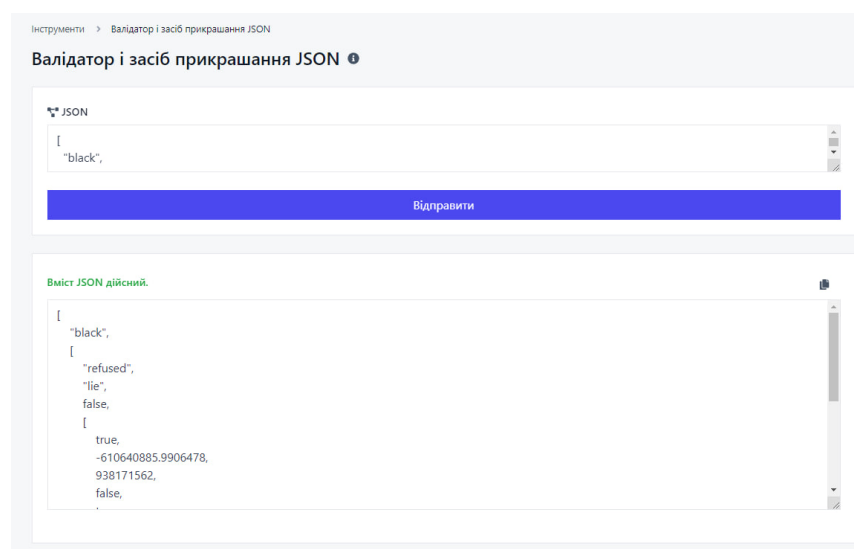


Рисунок 3.41 – Тестування інструменту «Валідатор і засіб прикрашання JSON»

Форматування/прикрашення SQL (рис. 3.42): Інструмент, який автоматично організовує код SQL-запитів, додаючи відступи, вирівнювання та форматування для полегшення читабельності. Це допомагає покращити структуру SQL-коду, зробити його більш зрозумілим та зручним для редагування та обслуговування.

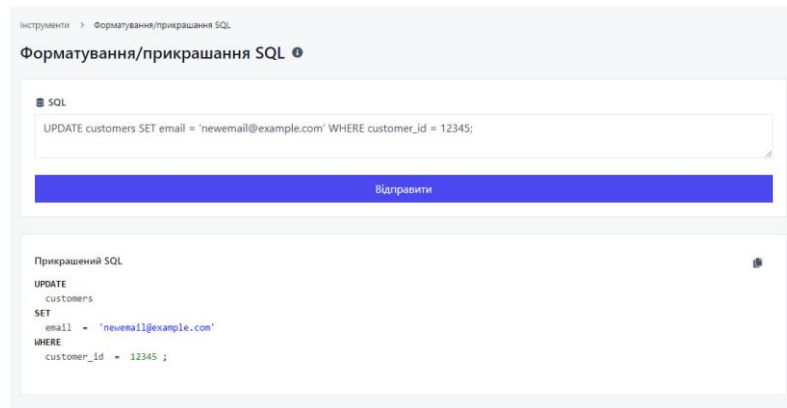


Рисунок 3.42 – Тестування інструменту «Форматування/прикрашення SQL»

Конвертер сутностей HTML (рис. 3.43): Інструмент, який перетворює спеціальні символи HTML (наприклад, "<", ">", "&") в їх відповідні сутності (наприклад, "<", ">", "&"), щоб уникнути конфліктів з розміткою та забезпечити коректне відображення HTML-коду у веб-браузерах.

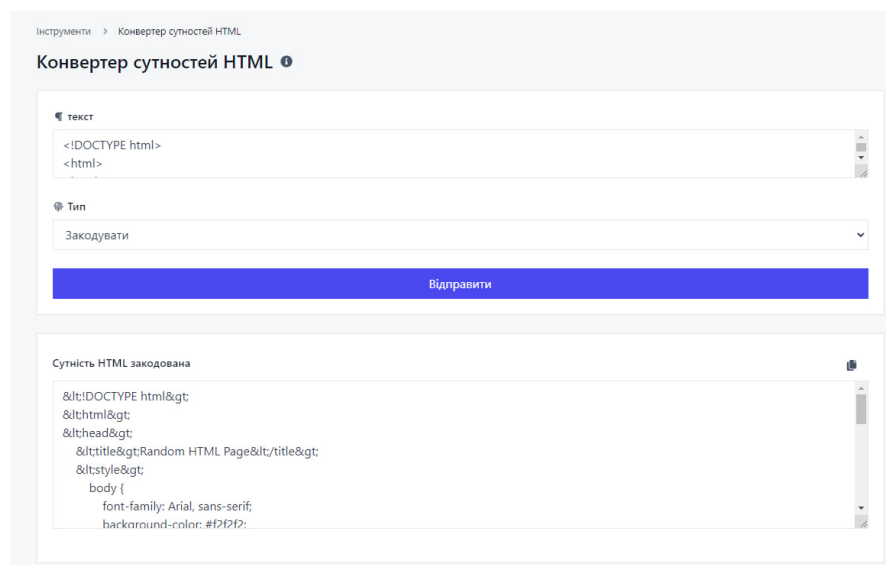


Рисунок 3.43 – Тестування інструменту «Конвертер сутностей HTML»

BBCode в HTML (рис. 3.44): Інструмент, який перетворює текст, написаний з використанням BBCode (мови розмітки для форумів), в відповідний HTML-код. Це дозволяє правильно відображати форматування, зображення, посилання та інші елементи, що використовуються в BBCode, в HTML-структурі, що використовується на веб-сторінках.

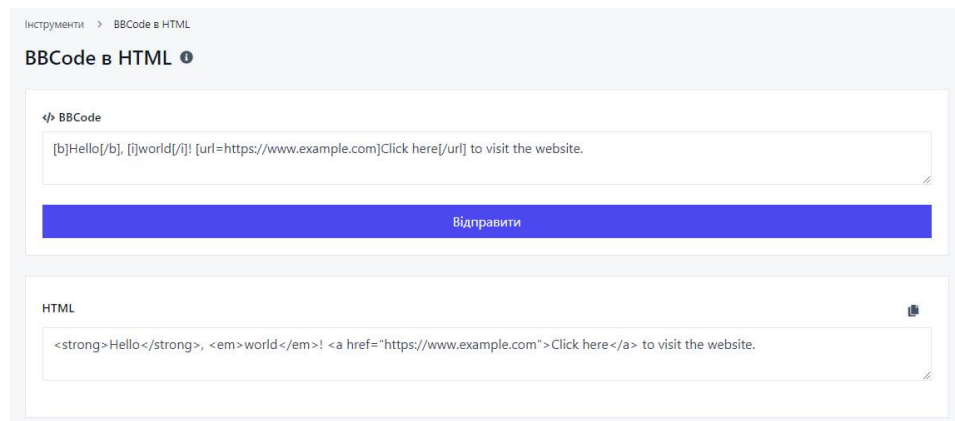


Рисунок 3.44 – Тестування інструменту «BBCode в HTML»

Markdown в HTML (рис. 3.45): Інструмент, який перетворює текст, написаний з використанням синтаксису Markdown, в HTML-код. Це дозволяє відображати правильну розмітку, стилізацію, посилання та інші елементи, що використовуються в Markdown, у веб-браузерах за допомогою HTML-структури.

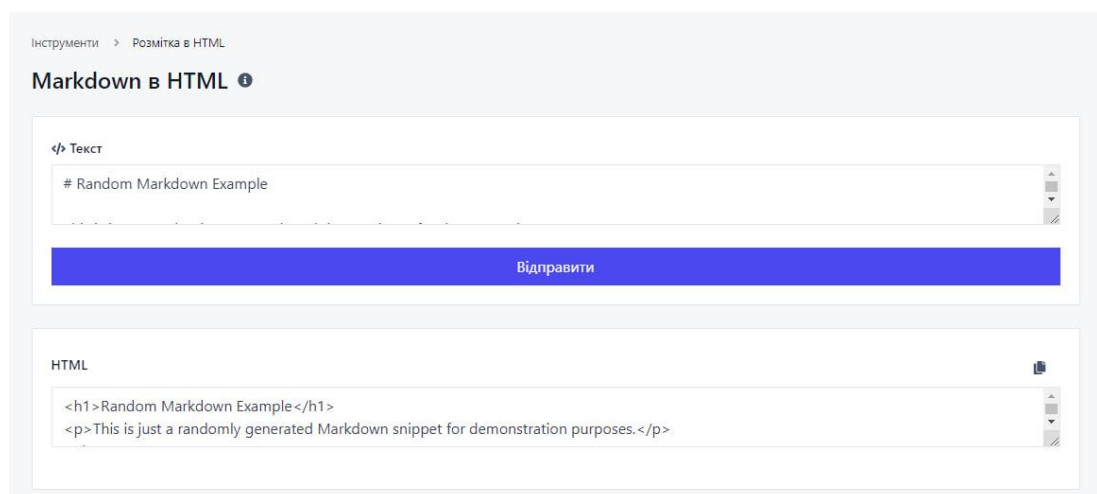


Рисунок 3.45 – Тестування інструменту «BBCode в HTML»

Парсер агента користувача (рис. 3.46): Інструмент, який аналізує рядок агента користувача (User Agent) в HTTP-запиті веб-браузера і витягує з нього інформацію про браузер, операційну систему, пристрій, мову та інші характеристики. Це допомагає визначити тип пристрою, з якого здійснюється запит, і адаптувати веб-сторінку або веб-додаток до потреб конкретного користувача.

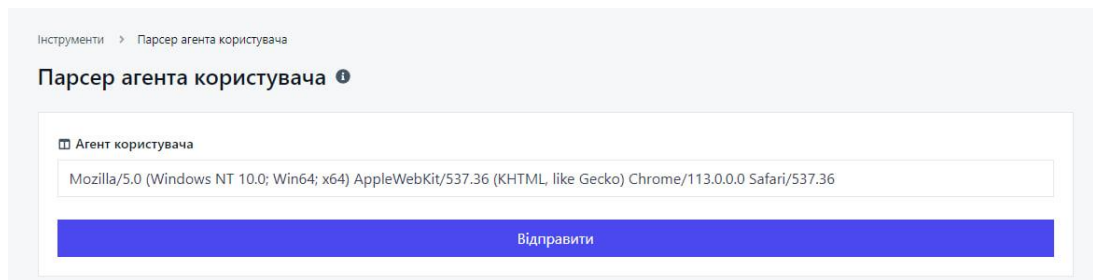


Рисунок 3.46 – Тестування інструменту «Парсер агента користувача»

Парсер URL-адрес (рис. 3.47): Інструмент, який аналізує URL-адресу і розбиває її на складові частини, такі як протокол, доменне ім'я, порт, шлях, параметри та інші. Це дозволяє отримати доступ до окремих елементів URL-адреси для подальшої обробки, перебудови або виконання відповідних дій на основі отриманих даних.

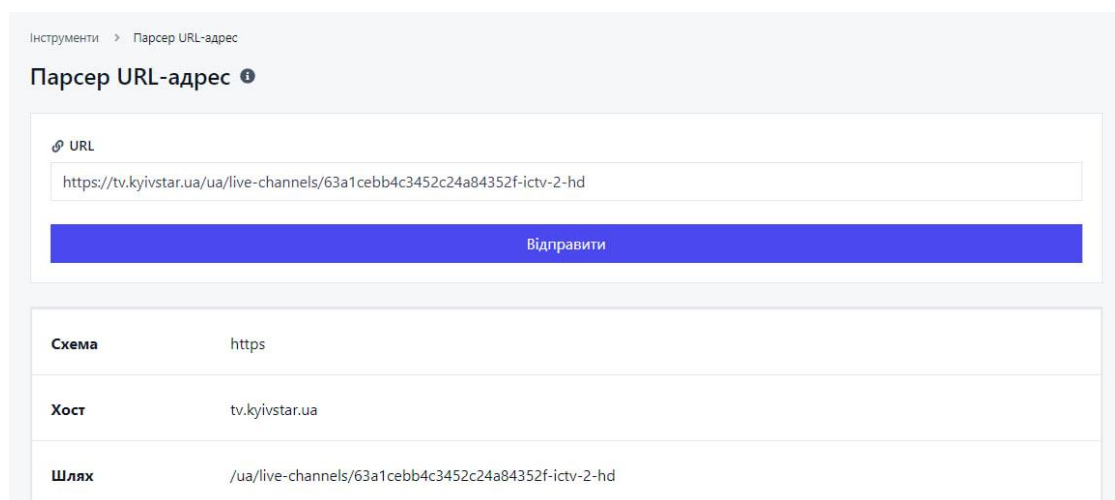


Рисунок 3.47 – Тестування інструменту «Парсер URL-адрес»

Окрім функціонального тестування інструментів веб-розробника, було проведено тестування безпеки системи. Було перевірено, чи належним чином захищені та зашифровані дані користувачів, чи присутні вразливості у механізмах авторизації, чи забезпечується конфіденційність даних (рис. 3.48).

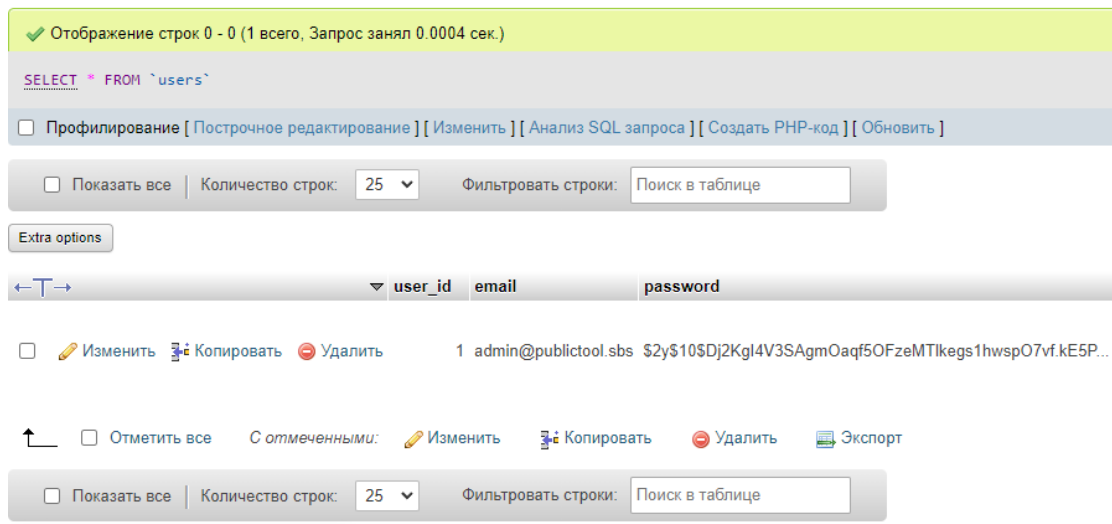


Рисунок 3.48 – Зашифровані дані в таблиці з даними користувачів

Після завершення тестування, були зібрані та проаналізовані дані про продуктивність системи, час відгуку та виконання операцій. Були виявлені та виправлені проблеми та помилки, щоб забезпечити якість та надійність системи.

В результаті роботи над «Інформаційною системою каталогізації інструментів веб-розробника» було підтверджено, що система працює коректно, функціонально та забезпечує безпеку даних користувачів.

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи було розроблено та реалізовано «Інформаційну систему каталогізації інструментів веб-розробника». Система покликана полегшити процес вибору, пошуку та використання інструментів для веб-розробників, підвищити їхню ефективність та якість роботи, а також зменшити витрати часу та ресурсів на рутинні задачі.

Аналіз потреб веб-розробників допоміг визначити ключові функціональні можливості, які повинна містити система. Цей аналіз став основою для проектування архітектури системи, розробки інтерфейсу та бази даних.

«Інформаційна система каталогізації інструментів веб-розробника» була створена за допомоги мов програмування HTML, CSS, JavaScript, PHP та SQL, з використанням СУБД MySQL для бази даних. Фреймворк Bootstrap був використаний для створення привабливого та зручного для користувача інтерфейсу. Цей набір технологій забезпечив ефективність, адаптивність та надійність системи для кінцевого користувача.

Основною новаторською рисою розробленої системи є об'єднання різноманітних інструментів веб-розробки в одній системі. Це забезпечує їх легке використання та сумісність, а також актуальність системи в динамічному світі веб-розробки.

Розроблена інформаційна система надає веб-розробникам зручний інструментарій для швидкого доступу до актуальних інструментів. Виконана робота може слугувати основою для подальших досліджень та розробок в даній області, спрямованих на подальше підвищення ефективності роботи веб-розробників.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Свєрчков В. ЩО ПОВИНЕН ЗНАТИ FRONTEND РОЗРОБНИК У 2023 РОЦІ. *itvdn.com*. URL: <https://itvdn.com/ua/blog/article/frontend-2023> (Дата звернення: 31.05.2023).
2. How Developers Really Spend Their Time. *SweetCode.io*. URL: <https://sweetcode.io/developers-really-spend-time-hint-not-coding/> (Дата звернення: 31.05.2023).
3. Pyzhianov A. How much time do web developers spend on searching information during work?. *Quora*. URL: <https://www.quora.com/How-much-time-do-web-developers-spend-on-searching-information-during-work> (Дата звернення: 31.05.2023).
4. Developers Spend 10-25% of Their Time Fixing Bugs in the Finished Product. *SPINCAREER*. URL: <https://spincareer.com/developers-spend-10-25-of-their-time-fixing-bugs-in-the-finished-product/> (Дата звернення: 31.05.2023).
5. Girba T. Developers spend most of their time figuring the system out. *feenk blog*. URL: <https://lepiter.io/feenk/developers-spend-most-of-their-time-figuring-the-system-out/> (Дата звернення: 31.05.2023).
6. Nuñez M. Developers embrace AI Tools but face ‘Big Code’ challenges, survey finds. *VentureBeat*. URL: <https://venturebeat.com/ai/developers-embrace-ai-tools-but-face-big-code-challenges-survey-finds/> (Дата звернення: 31.05.2023).
7. Kalliamvakou E. Research: quantifying GitHub Copilot’s impact on developer productivity and happiness. *GitHub Blog*. URL: <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/> (Дата звернення: 31.05.2023).
8. HTML - Вікіпедія. *Wikipedia*. URL: <https://uk.wikipedia.org/wiki/HTML> (Дата звернення: 31.05.2023).
9. Лютий С. Що таке CSS?. *BlogChain*. URL: <https://blogchain.com.ua/shcho-take-css/> (Дата звернення: 31.05.2023).

10. JavaScript - Клієнтською мовою програмування, що робить сторінки сайту інтерактивними. *AstwellSoft*. URL: <https://astwellsoft.com/uk/blog/tehnology/javascript.html> (Дата звернення: 31.05.2023).
11. Що таке PHP?. *W3schoolsUA*. URL: <https://w3schoolsua.github.io/php/index.html#gsc.tab=0> (Дата звернення: 31.05.2023).
12. Де використовується SQL і чому він так потрібен програмістам. *IT STEP Academy*. URL: <https://kiev.itstep.org/blog/where-sql-is-used-and-why-programmers-need-it-so-much> (Дата звернення: 31.05.2023).
13. Знайомство з Visual Studio Code. *Romul Name*. URL: <https://romul.name/blog/znayomstvo-z-visual-studio-code/> (Дата звернення: 31.05.2023).
14. 9 актуальних CSS-фреймворків для Front-end розробника. *Web-Acedemy.UA*. URL: <https://web-academy.ua/blog/junior/junior-9-css-frameworks-for-front-end> (Дата звернення: 31.05.2023).
15. Колчак Н. Bootstrap: що це, з чого почати вивчення і як використовувати. *Друзі*. URL: <https://druzy.com.ua/bootstrap-sho-ce-z-chogo-rochati-vivchennia-i-iak-vikoristovuyati/> (Дата звернення: 31.05.2023).
16. Rud A. Для чого використовуються COOKIES і навіщо їх чистити?. *HyperHost.UA*. URL: <https://hyperhost.ua/info/uk/dlya-chogo-vikoristovuyutsya-cookies-i-navishcho-ikh-chistiti> (Дата звернення: 31.05.2023).
17. ТИПИ ФАЙЛІВ СОOKIE Й ІНШИХ ТЕХНОЛОГІЙ, ЯКІ ВИКОРИСТОВУЄ GOOGLE. *Google Policies*. URL: <https://policies.google.com/technologies/cookies?hl=uk> (Дата звернення: 31.05.2023).
18. MySQL - Вікіпедія. *Wikipedia*. URL: <https://uk.wikipedia.org/wiki/MySQL> (Дата звернення: 31.05.2023).

19. MySQL. Вікі-енциклопедія кафедри інформатики і ІКТ. URL: <https://wikifizmat.udpu.edu.ua/index.php?title=MySQL> (Дата звернення: 31.05.2023).

20. ER-діаграма - це Опис, види, правила побудови. *hi-news.pp.ua*. URL: <https://hi-news.pp.ua/kompyuteri/14668-er-dagrama-ce-opis-vidi-pravila-pobudovi.html> (Дата звернення: 31.05.2023).

21. Капча (CAPTCHA) – що це таке і навіщо потрібна. *Termin.in.ua*. URL: <https://termin.in.ua/kapcha-captcha/> (Дата звернення: 31.05.2023).

22. Запам'ятати мене. *LinkedIn*. URL: <https://www.linkedin.com/help/linkedin/answer/a1342648/> (Дата звернення: 31.05.2023).

ДОДАТКИ

Додаток А. Лістинг програмного коду

```
<?php
namespace Tools;
use Tools\Security\Auth;
use Tools\Security\CSRFToken;
use Tools\Routing\Router;
class App
{
    Database\Database::initialize();
    Cache::initialize();
    Language::initialize();
    Router::parseurl();
    Router::parselanguage();
    Router::parsecontroller();
    $controller = Router::getcontroller(Router::$controller, Router::$path);
    $method = Router::parsemethod($controller);
    $params = Router::getparams();
    $settings = (new \Tools\Models\Settings()->get());
    \Tools\Settings::initialize($settings);
    Language::setDefaultbyname($settings
        ->main
        ->defaultlanguage);
    Title::initialize($settings
        ->main
        ->title);
    Meta::initialize();
    date_default_timezone_set(Date::$defaulttimezone);
```

```

Date::$timezone = datedefaulttimezoneget();
Date::$date = Date::get();
if (Auth::check())
{
    $user = Auth::$user;
    if (!$user)
    {
        Auth::logout();
    }
    if (!$user->lastactivity || (new \DateTime($user->lastactivity))
        ->modify('+5 minutes') < (new \DateTime()))
    {
        (new \Tools\Models\User())
            ->updateLastactivity(Auth::$user_id);
    }
    if (!isset($_GET['setlanguage']) && Language::$name != $user->language)
    {
        Language::setbyname($user->language);
    }
    if (isset($_GET['setlanguage']) && arraykeyexists($_GET['setlanguage'],
Language::$languages) && Language::$name != $user->language)
    {
        db()
            ->where('user_id', Auth::$user_id)->update('users', ['language' =>
$_GET['setlanguage']]);
        \Tools\Cache::$adapter->deleteItemsByTag('user_id=' . Auth::$user_id);
    }
    Date::$timezone = $user->timezone;
    Auth::$user = $user;

```

```

    }
else
{
    Auth::$user = new \StdClass();
}
CSRFTOKEN::set('token');
CSRFTOKEN::set('globaltoken');
if (Router::$languagecode == Language::$defaultcode)
{
    redirect(Router::$originalrequest);
}
$browserlanguage = isset($_SERVER['HTTP_ACCEPT_LANGUAGE']) ?
mb_substr($_SERVER['HTTP_ACCEPT_LANGUAGE'], 0, 2) : null;
if (settings()
    ->main->autolanguage &&
Router::$controllersettings['nobrowserlanguage'] == false &&
!Router::$languagecode && !Auth::check() && $browserlanguage &&
Language::$defaultcode != $browserlanguage && array_search($browserlanguage,
Language::$activelanguages))
{
    if (!isset($_SERVER['HTTP_REFERER']) ||
(isset($_SERVER['HTTP_REFERER']) &&
parseurl($_SERVER['HTTP_REFERER']) ['host'] != parseurl(SITE_URL) ['host']))
    {
        header('Location: ' . SITE_URL . $browserlanguage . '/' .
Router::$originalrequest);
    }
}
$controller->addparams(['params' => $params, 'user' => Auth::$user]);

```

```
if (!is_null(Router::$controllersettings['Auth']))
{
    Auth::guard(Router::$controllersettings['Auth']);
}
call_user_func_array([$controller, $method], []);
$controller->run();
Database\Database::close();
}
}
```