

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему РОЗРОБКА КАЛЬКУЛЯТОРА ВАРТОСТІ ПОСЛУГ ДОСТАВКИ

Виконав: студент(ка) 4 курсу, групи КНТ-522

Спеціальності 123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
«Комп'ютерна інженерія»

ХУДЯКОВ М.Д.
(прізвище та ініціали)

Керівник ГРУШКО С.С.
(прізвище та ініціали)

Рецензент ХАРИТОНОВ О.Б.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти бакалаврський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)

Освітня програма (спеціалізація) Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Завідувач кафедри КУДЕРМЕТОВ Р.К.

“ ” _____ 2026 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ХУДЯКОВА Максима Денисовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка калькулятора вартості послуг доставки
 керівник проєкту (роботи) к. т. н., доцент, ГРУШКО Світлана Сергіївна

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

затверджені наказом вищого навчального закладу від “14” квітня 2026 року № 160

2. Строк подання студентом проєкту (роботи) 30 травня 2026 року

3. Вихідні дані до проєкту (роботи) автономна односторінкова веб-система у вигляді єдиного HTML-файлу; вбудована тарифна база з підтримкою чотирьох зон доставки та трьох типів відправлень; адаптивний інтерфейс з підтримкою мобільних пристроїв, планшетів та настільних комп'ютерів; україномовний інтерфейс; розрахунок вартості на стороні клієнта без серверної інфраструктури; робота без підключення до мережі Інтернет після початкового завантаження

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- 1) Аналітичний огляд предметної області;
- 2) Проектування веб-системи;
- 3) Реалізація веб-системи;
- 4) Апаратне забезпечення системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ГРУШКО С. С., к.т.н., доцент		
нормоконтроль	ГОЛУБ Т.В., к.т.н., доцент		

7. Дата видачі завдання 14.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технічного завдання. Огляд ринку послуг доставки в Україні та існуючих калькуляторів вартості доставки	14.04.26 – 20.04.26	
2	Дослідження технологій розробки односторінкових веб-застосунків	20.04.26 - 25.04.26	
3	Проектування архітектури веб-системи, алгоритму розрахунку вартості та структури тарифних даних	25.04.26 - 03.05.26	
4	Розробка HTML-розмітки, CSS-стилізації та адаптивного дизайну інтерфейсу калькулятора	03.05.26 - 10.05.26	
5	Програмування бізнес-логіки розрахунку вартості доставки та валідації вхідних даних	10.05.26 – 20.05.26	
6	Тестування та налагодження додатку на різних пристроях і браузерах	20.05.26 - 22.05.26	
7	Оформлення пояснювальної записки	22.05.26 - 25.05.26	
8	Оформлення графічного та допоміжного матеріалу	25.05.26 – 30.05.26	

Студент (ка)

Максим ХУДЯКОВ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Світлана ГРУШКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи бакалавра: 70 с., 3 табл., 5 рис., 1 додаток, 25 джерел.

КАЛЬКУЛЯТОР ВАРТОСТІ ДОСТАВКИ, ВЕБ-СИСТЕМА, ОДНОСТОРІНКОВИЙ ЗАСТОСУНОК, HTML5, CSS3, JAVASCRIPT, SPA, ЛОГІСТИКА, ТАРИФНА ЗОНА

Об'єкт розробки – односторінковий веб-застосунок для розрахунку вартості послуг доставки.

Мета роботи – розробка автономної односторінкової веб-системи калькулятора вартості послуг доставки, що забезпечує зручний розрахунок вартості відправлення з урахуванням тарифних зон, типу доставки та додаткових послуг для підприємців малого та середнього бізнесу.

Розроблено повнофункціональний односторінковий веб-застосунок калькулятора вартості доставки, реалізований у вигляді єдиного HTML-файлу обсягом до 50 КБ. Алгоритм розрахунку враховує п'ять складових: базовий тариф тарифної зони, надбавку за розрахункову вагу, коефіцієнт типу доставки, коефіцієнт пункту призначення та вартість додаткових послуг. Створено адаптивний інтерфейс із використанням технологій CSS Grid Layout, Flexbox та медіа-запитів, що забезпечує коректне відображення на пристроях із шириною екрана від 320 до 1440 і більше пікселів.

ABSTRACT

Explanatory note to the bachelor's degree thesis: 70 p., 3 tables, 5 fig., 1 appendix, 25 sources.

DELIVERY COST CALCULATOR, WEB SYSTEM, SINGLE PAGE APPLICATION, HTML5, CSS3, JAVASCRIPT, SPA, LOGISTICS, TARIFF ZONE

Object of development – single-page web application for calculating delivery service costs.

Objective of the work – development of an autonomous single-page web system of a delivery cost calculator that provides convenient calculation of shipment costs taking into account tariff zones, delivery types, and additional services for small and medium-sized business entrepreneurs.

A fully functional single-page delivery cost calculator web application has been developed, implemented as a single HTML file with a size of up to 50 KB. The calculation algorithm considers five components: base tariff of the tariff zone, surcharge for chargeable weight, delivery type coefficient, destination coefficient, and additional services cost. A responsive interface using CSS Grid Layout, Flexbox, and media queries has been implemented, ensuring correct display on devices with screen widths from 320 to 1440 and more pixels.

ЗМІСТ

Вступ.....	8
1 Аналітичний огляд предметної області	10
1.1 Аналіз ринку послуг доставки в Україні.....	10
1.2 Огляд наявних рішень калькуляторів доставки	13
1.3 Постановка задачі та формування вимог до системи	17
1.4 Висновки до розділу 1	21
2 Проєктування веб-системи.....	22
2.1 Вибір технологій розробки.....	22
2.1.1 Обґрунтування вибору HTML5, CSS3, JavaScript	22
2.1.2 Архітектура односторінкового застосунку (SPA)	24
2.2 Проєктування інтерфейсу користувача.....	26
2.2.1 Макети екранних форм.....	26
2.2.2 UX/UI принципи побудови інтерфейсу	27
2.3 Проєктування логіки розрахунку вартості.....	29
2.3.1 Алгоритм калькуляції	29
2.3.2 Діаграма послідовності обчислень	30
2.3.3 Структура даних (тарифи, зони доставки)	32
2.4 Структура HTML-документа.....	33
2.4.1 Семантична розмітка	33
2.4.2 Модульна організація коду	34
2.5 Висновки до розділу 2	35
3 Реалізація веб-системи.....	36
3.1 Розробка інтерфейсної частини	36

	7
3.2 Реалізація бізнес-логіки	43
3.3 Результат розрахунку та відображення даних	47
3.4 Тестування системи	49
3.5 Висновки до розділу 3	53
4 Апаратне забезпечення системи	54
4.1 Вимоги до серверного обладнання.....	54
4.2 Вимоги до клієнтського обладнання.....	56
4.3 Вимоги до мережевої інфраструктури	57
4.4 Висновки до розділу 4	58
Висновки	59
Перелік джерел посилань	62
Додаток А Слайди презентації.....	65

ВСТУП

Стрімкий розвиток електронної комерції в Україні та світі кардинально змінив підходи до організації торгівлі та логістики. Згідно з даними дослідницької компанії Statista, глобальний ринок електронної комерції у 2024 році досяг позначки 6,3 трильйона доларів США, демонструючи щорічний приріст на рівні 8,76% порівняно з попереднім періодом. Український сегмент e-commerce також показує значну динаміку: за результатами 2024 року загальна сума товарів та послуг, придбаних в інтернеті, склала 239 мільярдів гривень, що на 25% перевищує показники 2023 року [1]. Таке зростання обсягів онлайн-торгівлі безпосередньо корелює з підвищенням попиту на послуги доставки, що актуалізує потребу в ефективних інструментах розрахунку їх вартості.

Сучасний споживач очікує від інтернет-магазинів не лише широкого асортименту товарів, але й прозорої цінової політики щодо логістичних послуг. Калькулятор вартості доставки стає невід'ємним елементом користувацького інтерфейсу будь-якої торговельної платформи, оскільки дозволяє покупцеві завчасно оцінити повну вартість замовлення. Відсутність такого інструменту або його некоректна робота призводить до відмови від покупок на етапі оформлення замовлення, що підтверджується численними маркетинговими дослідженнями [2].

Актуальність теми дослідження обумовлена кількома ключовими факторами. По-перше, український ринок поштових та кур'єрських послуг характеризується високою конкуренцією: за даними компанії «Нова пошта», у 2024 році було оброблено понад 480 мільйонів відправлень, що на 16% більше порівняно з 2023 роком [3]. По-друге, тарифна політика логістичних операторів постійно змінюється під впливом економічних факторів, зокрема коливань цін на паливо та електроенергію. По-третє, впровадження цифрових технологій дозволяє автоматизувати процес розрахунку вартості доставки, мінімізуючи людський фактор та підвищуючи точність калькуляцій.

Особливої актуальності набуває розробка веб-орієнтованих калькуляторів доставки, реалізованих у форматі односторінкових застосунків (Single Page Application, SPA). Такий підхід забезпечує миттєву реакцію інтерфейсу на дії користувача, оскільки всі необхідні ресурси завантажуються одноразово при відкритті сторінки, а подальша взаємодія відбувається без перезавантаження документа [4]. Односторінкова архітектура значно спрощує процес розгортання та підтримки веб-системи, що є важливим критерієм для малого та середнього бізнесу з обмеженими ІТ-ресурсами.

1 АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз ринку послуг доставки в Україні

Ринок логістичних послуг України є одним із найбільш динамічних секторів національної економіки, що демонструє стабільне зростання навіть в умовах економічної нестабільності та воєнного стану. Розвиток електронної комерції став основним драйвером цього зростання, оскільки онлайн-покупки неможливі без ефективної системи доставки товарів кінцевому споживачеві. За даними дослідження групи компаній EVO, частка електронної комерції в загальному обсязі роздрібної торгівлі України у 2024 році склала приблизно 10%, що вдвічі менше, ніж у Польщі та США, проте кількість інтернет-покупців та частота покупок демонструють стійку тенденцію до зростання [5].

Станом на 2024 рік в Україні налічується близько 11 мільйонів активних інтернет-покупців, які в середньому здійснюють 17 покупок онлайн на рік. Така активність генерує значний потік відправлень, який обслуговується розгалуженою мережею поштових та кур'єрських компаній. Структура ринку послуг доставки характеризується наявністю одного домінуючого гравця – компанії «Нова пошта», яка контролює понад 60% ринку експрес-доставки, а також низки менших операторів, включаючи державну «Укрпошту», приватну «Meest Пошта» та інших учасників [6].

Компанія «Нова пошта» є безумовним лідером українського ринку логістичних послуг. За підсумками 2024 року компанія встановила рекорд із доставки посилок, виконавши 480 мільйонів відправлень, що на 16% більше порівняно з 2023 роком, коли було перевезено 412 мільйонів посилок [7]. Така динаміка свідчить про високий рівень довіри споживачів та ефективність бізнес-моделі компанії. Мережа «Нової пошти» станом на кінець 2024 року налічувала понад 37 тисяч точок обслуговування, включаючи 14 336 відділень та 30 647 поштоматів, що забезпечує максимальне охоплення території країни.

Фінансові показники компанії також демонструють позитивну динаміку: консолідований дохід «Нової пошти» у 2024 році зріс на 22% і склав 44,77 млрд грн, хоча чистий прибуток зменшився на 36,9% до 2,5 млрд грн, що пояснюється значними інвестиціями в розвиток інфраструктури – понад 7,3 млрд грн було вкладено в інновації, нові технології та послуги [8]. Компанія активно розширює міжнародну присутність, здійснюючи доставку в 16 країнах Європи, а обсяг міжнародних відправлень у 2024 році зріс на 86% порівняно з попереднім роком.

Державна компанія «Укрпошта» займає другу позицію на ринку поштових послуг, проте значно поступається приватним операторам за обсягами експрес-доставки. Основною конкурентною перевагою «Укрпошти» є найнижчі базові тарифи серед усіх учасників ринку, що обумовлено частковим державним субсидуванням діяльності компанії. Згідно з відкритими даними, лише у 2022 році «Укрпошта» отримала понад 3 млрд грн від держави у вигляді різноманітних компенсацій і дотацій [9]. Станом на 2024 рік базові тарифи «Укрпошти» на пересилку у межах області становлять: до 2 кг – 40 грн, до 5 кг – 55 грн, до 10 кг – 70 грн, що суттєво нижче за тарифи приватних операторів.

Компанія «Meest Пошта» спеціалізується переважно на міжнародних відправленнях та займає значну частку ринку доставки посилок з-за кордону. Окрім цього, компанія розвиває внутрішню мережу відділень та поштоматів, пропонуючи конкурентні тарифи на доставку по Україні. «Meest Пошта» активно співпрацює з маркетплейсами та інтернет-магазинами, забезпечуючи інтеграцію своїх послуг у процес оформлення замовлень.

Аналіз факторів, що впливають на вартість послуг доставки, дозволяє виділити кілька ключових параметрів. Перш за все, це фізичні характеристики відправлення: фактична вага та габаритні розміри. Логістичні компанії застосовують концепцію об'ємної (або габаритної) ваги, яка розраховується за формулою: $V = (D \times W \times H) / K$, де D , W , H — довжина, ширина та висота посилки в сантиметрах, а K — коефіцієнт, який для більшості перевізників становить 4000 [10]. При тарифікації враховується більше значення з двох: фактична або об'ємна

вага. Такий підхід дозволяє компаніям оптимізувати завантаження транспортних засобів та справедливо розподіляти логістичні витрати.

Другим важливим фактором є відстань доставки або належність населених пунктів відправлення та отримання до певних тарифних зон. Більшість операторів використовують зональну систему тарифікації, де вартість залежить від типу маршруту: в межах одного міста, у межах однієї області, між різними областями, доставка в сільську місцевість. Зокрема, «Нова пошта» застосовує доплату в розмірі 25 гривень за доставку у села та селища понад базовий тариф [11].

Тип доставки також суттєво впливає на кінцеву вартість послуги. Можна виділити такі основні варіанти: доставка «відділення — відділення» (найдешевший варіант), «відділення — поштомот», «відділення — адреса» (кур'єрська доставка), «адреса — відділення» (забір кур'єром з адреси). Кур'єрська доставка традиційно коштує дорожче за самовивіз із відділення, проте забезпечує максимальну зручність для отримувача [12].

Додаткові послуги формують окрему категорію витрат. До них належать: декларована вартість відправлення (страхування), післяплата (грошовий переказ від отримувача до відправника), пакування, зворотна доставка документів, експрес-доставка тощо. Наприклад, послуга накладеного платежу у «Новій пошті» тарифікується як 2% від суми переказу, але не менше 20 гривень [13]. Такі додаткові послуги можуть суттєво збільшити загальну вартість відправлення, особливо для товарів із високою оціночною вартістю.

Окремої уваги заслуговує аналіз сезонності попиту на логістичні послуги. Традиційно пікове навантаження припадає на передноворічний період: за даними «Нової пошти», 22 грудня 2025 року було опрацьовано рекордні 2,41 млн відправлень за добу, що на 7,1% перевищило попередній рекорд [14]. Така сезонність створює додаткові виклики для логістичних операторів у частині забезпечення стабільної якості послуг та дотримання заявлених термінів доставки.

Війна, що триває з 2022 року, суттєво вплинула на функціонування ринку логістичних послуг. Проблеми з логістикою, безпекою та доступом до окремих регіонів ускладнили процес доставки та призвели до збільшення операційних

витрат. Водночас, компанії продемонстрували високу адаптивність, змінюючи маршрути доставки та розширюючи роботу в регіонах, що не постраждали від бойових дій [15]. Парадоксально, але саме воєнний час спричинив додатковий поштовх до розвитку онлайн-торгівлі, оскільки частина населення почала надавати перевагу дистанційним покупкам із міркувань безпеки.

Важливим трендом останніх років є прихід на український ринок міжнародних гравців електронної комерції. Зокрема, у 2024 році агресивну маркетингову кампанію розгорнув китайський маркетплейс Temu, який пропонує товари за низькими цінами з безкоштовною доставкою. Аналітики відзначають, що такі гравці здатні суттєво змінити конкурентне середовище та вплинути на тарифну політику вітчизняних логістичних операторів [16].

Таким чином, український ринок послуг доставки характеризується високим рівнем конкуренції, значними обсягами відправлень та динамічною тарифною політикою. Для споживачів та підприємців актуальною залишається потреба в зручних інструментах розрахунку вартості доставки, які б враховували всі перелічені фактори та надавали точну інформацію в режимі реального часу.

1.2 Огляд наявних рішень калькуляторів доставки

Сучасні логістичні компанії та інтернет-магазини активно використовують калькулятори вартості доставки як важливий інструмент взаємодії з клієнтами. Аналіз існуючих рішень дозволяє виявити типові підходи до проєктування таких систем, їх функціональні можливості, переваги та недоліки. У цьому підрозділі буде розглянуто калькулятори провідних українських логістичних операторів, а також загальні принципи побудови подібних веб-інструментів.

Калькулятор вартості доставки «Нової пошти» є найбільш відомим та найчастіше використовуваним в Україні. Система пропонує користувачеві вказати місто відправлення та отримання, тип доставки (відділення, поштомат, адреса),

параметри відправлення (вага, габарити), а також додаткові послуги. Інтерфейс калькулятора інтуїтивно зрозумілий, підтримує автозаповнення назв населених пунктів та відділень. Перевагами рішення є висока точність розрахунків, інтеграція з актуальною тарифною базою компанії та можливість одразу створити електронну накладну на основі розрахованих параметрів [17].

Технічно калькулятор «Нової пошти» реалізовано як клієнт-серверний застосунок, де клієнтська частина звертається до API компанії для отримання актуальних тарифів та довідкової інформації. Такий підхід забезпечує централізоване оновлення даних та гарантує, що користувач завжди отримує актуальну інформацію. Водночас, залежність від серверного API означає, що калькулятор не функціонуватиме без доступу до інтернету або у разі технічних проблем на стороні провайдера послуг [18].

Калькулятор «Укрпошти» доступний за адресою calc.ukrposhta.ua та надає можливість розрахувати вартість внутрішніх та міжнародних відправлень. Функціональність системи дещо обмежена порівняно з приватними конкурентами: відсутня можливість одразу оформити відправлення, менш зручний механізм вибору населених пунктів. Проте калькулятор коректно відображає всі тарифи державного оператора, включаючи пільгові умови для окремих категорій відправлень [19].

Компанія «Meest Пошта» пропонує калькулятор із розширеними можливостями для міжнародних відправлень. Користувач може розрахувати вартість доставки в понад 70 країн світу, при цьому система автоматично інформує про митні обмеження, максимальну допустиму вагу та габарити для кожного напрямку. Калькулятор також враховує особливості оподаткування посилок залежно від країни призначення та оціночної вартості вкладення [20]. Формула розрахунку об'ємної ваги для «Meest Пошта» стандартна: $(\text{Довжина} \times \text{Ширина} \times \text{Висота}) / 4000$, що відповідає практиці більшості міжнародних перевізників.

Окрім калькуляторів логістичних операторів, на ринку представлені агрегатори, що дозволяють порівняти вартість доставки різними компаніями. Такі сервіси збирають інформацію з відкритих джерел або використовують офіційні API

партнерів для формування порівняльних таблиць. Перевагою агрегаторів є можливість вибору оптимального варіанту за критеріями ціни, швидкості доставки або зручності отримання. Водночас, актуальність даних у таких системах може відставати від офіційних тарифів операторів [21].

Транспортні компанії, що спеціалізуються на вантажних перевезеннях, пропонують калькулятори з розширеним функціоналом для логістичного планування. Наприклад, калькулятор компанії SAT дозволяє вказати не лише параметри вантажу, але й проміжні точки маршруту, спеціальні умови перевезення, потребу в додаткових послугах (експедирування, охорона, пакування) [22]. Такі системи орієнтовані на професійних користувачів — логістів та менеджерів з постачання, тому їхній інтерфейс може бути надмірно складним для пересічного споживача.

Аналіз технологічних рішень, що застосовуються у існуючих калькуляторах доставки, дозволяє виділити кілька підходів до їх реалізації. Перший підхід передбачає використання серверних технологій (PHP, Python, Node.js) для обробки запитів та формування відповідей. Калькулятор відправляє дані на сервер, де відбувається розрахунок вартості на основі актуальної тарифної бази, після чого результат повертається користувачеві. Такий підхід забезпечує безпеку бізнес-логіки та можливість централізованого оновлення тарифів, проте створює додаткове навантаження на сервер та залежність від якості інтернет-з'єднання [23].

Другий підхід базується на клієнтських технологіях, де вся логіка розрахунку виконується безпосередньо в браузері користувача за допомогою JavaScript. Тарифні дані можуть бути вбудовані в код сторінки або завантажені у вигляді JSON-файлу. Перевагами такого підходу є миттєва реакція інтерфейсу на зміни вхідних параметрів, можливість роботи в режимі офлайн (після початкового завантаження), зменшення навантаження на сервер. Недоліком є потреба в ручному оновленні тарифів та потенційна вразливість бізнес-логіки для реверс-інжинірингу [24].

Гібридний підхід поєднує переваги обох попередніх рішень: базова логіка розрахунку виконується на клієнті, тоді як актуальні тарифи та довідкові дані

періодично синхронізуються з сервера. Такий підхід застосовується у прогресивних веб-застосунках (PWA), які здатні кешувати дані та забезпечувати обмежену функціональність навіть без підключення до мережі [25].

З точки зору користувацького досвіду (UX), сучасні калькулятори доставки мають відповідати кільком базовим вимогам. По-перше, інтерфейс повинен бути інтуїтивно зрозумілим, щоб користувач міг виконати розрахунок без попереднього навчання. По-друге, система має надавати миттєвий зворотний зв'язок на дії користувача: підсвічувати помилки введення, показувати підказки, демонструвати проміжні результати. По-третє, калькулятор повинен коректно працювати на різних пристроях — від настільних комп'ютерів до мобільних телефонів, що вимагає застосування принципів адаптивного дизайну [26].

Дизайн користувацького інтерфейсу (UI) калькуляторів доставки традиційно будується за формовим принципом: користувач послідовно заповнює поля форми, після чого отримує результат розрахунку. Сучасні тенденції UI-дизайну передбачають використання «плоского» візуального стилю, обмеженої кольорової палітри, достатніх відступів між елементами для забезпечення зручності на сенсорних екранах. Важливим аспектом є візуальна ієрархія: ключові елементи (кнопка розрахунку, підсумкова вартість) повинні бути візуально виділені та легко впізнаватися [27].

Серед типових недоліків існуючих калькуляторів можна виділити такі: обмежені можливості кастомізації під потреби конкретного бізнесу; відсутність інтеграції з популярними CMS та платформами електронної комерції; складність вбудовування калькулятора на сторонні веб-ресурси; недостатня документованість API для розробників [28]. Ці недоліки створюють простір для розробки альтернативних рішень, орієнтованих на специфічні потреби малого та середнього бізнесу.

Окремої уваги заслуговує концепція односторінкового застосунку (Single Page Application, SPA) як архітектурного підходу до побудови калькуляторів. SPA характеризується тим, що всі необхідні ресурси (HTML, CSS, JavaScript) завантажуються одноразово при відкритті сторінки, а подальша взаємодія з

користувачем відбувається без перезавантаження документа [29]. Динамічне оновлення контенту здійснюється за допомогою JavaScript, який модифікує DOM-дерево сторінки у відповідь на дії користувача. Такий підхід забезпечує плавний та швидкий користувацький досвід, що особливо важливо для інтерактивних інструментів, таких як калькулятори.

Переваги SPA-підходу для реалізації калькулятора доставки включають: миттєву реакцію інтерфейсу на зміни вхідних даних; можливість реалізації складної інтерактивної логіки без звернень до сервера; спрощену процедуру розгортання (достатньо розмістити один HTML-файл на будь-якому веб-сервері); зменшення споживання трафіку за рахунок відсутності повторних завантажень сторінки [30]. Недоліки SPA традиційно пов'язують із SEO-оптимізацією та початковим часом завантаження, проте для інструментів типу калькулятора ці фактори мають меншу значущість.

Підсумовуючи огляд існуючих рішень, можна констатувати, що ринок калькуляторів вартості доставки представлений переважно пропрієтарними продуктами великих логістичних операторів, орієнтованими на їхні власні тарифи та послуги. Існує потреба в універсальних рішеннях, які б дозволяли підприємцям швидко налаштувати калькулятор під власні умови співпраці з перевізниками, не залежачи від зовнішніх API та складної серверної інфраструктури.

1.3 Постановка задачі та формування вимог до системи

На основі проведеного аналізу ринку послуг доставки та наявних програмних рішень сформульовано постановку задачі дипломної роботи: розробити веб-систему калькулятора вартості послуг доставки у вигляді односторінкового HTML-файлу, який може бути легко інтегрований у будь-який веб-ресурс або використаний автономно. Система повинна забезпечувати розрахунок вартості доставки на основі параметрів відправлення та обраних опцій, при цьому вся логіка

обробки даних має виконуватися на стороні клієнта без потреби у серверній інфраструктурі.

Функціональні вимоги до системи визначають перелік операцій, які вона повинна виконувати для задоволення потреб кінцевих користувачів:

- система повинна надавати форму для введення фізичних характеристик посилки (вага, довжина, ширина, висота), що дозволить розрахувати як фактичну, так і об'ємну вагу;
- користувач повинен мати можливість обрати спосіб доставки (відділення – відділення, кур'єрська доставка, поштомат) із відповідною зміною тарифу;
- система має враховувати різницю в тарифах залежно від напрямку (в межах міста, між містами, у сільську місцевість) ;
- калькулятор повинен підтримувати опції страхування вантажу (декларована вартість), накладеного платежу, пакування тощо з автоматичним перерахунком загальної вартості;
- результат калькуляції має оновлюватися в режимі реального часу при зміні будь-якого вхідного параметра без потреби натискати кнопку «Розрахувати»;
- система повинна відображати не лише підсумкову вартість, але й її складові (базовий тариф, доплати за додаткові послуги, комісії);
- калькулятор має перевіряти коректність введених значень та інформувати користувача про помилки (від'ємна вага, перевищення максимальних габаритів тощо).

Нефункціональні вимоги визначають якісні характеристики системи, що впливають на користувацький досвід та технічну підтримку:

- час реакції інтерфейсу на дії користувача не повинен перевищувати 100 мілісекунд, що забезпечує відчуття миттєвого відгуку системи;
- система повинна коректно функціонувати в усіх сучасних браузерях (Chrome, Firefox, Safari, Edge) останніх двох мажорних версій;

- інтерфейс калькулятора має коректно відображатися на екранах різних розмірів — від мобільних телефонів (ширина 320 пікселів) до настільних моніторів (Full HD та вище);
- система повинна функціонувати повністю на стороні клієнта, не вимагаючи серверної обробки запитів (за винятком початкового завантаження HTML-файлу);
- весь код системи (HTML, CSS, JavaScript) має міститися в одному файлі для спрощення розгортання та підтримки;
- інтерфейс має відповідати базовим принципам веб-доступності (контрастність кольорів, можливість навігації з клавіатури, семантична розмітка);
- структура коду повинна передбачати можливість модифікації тарифів та додавання нових опцій без значних змін в архітектурі системи.

Обмеження проєкту визначають рамки, в яких здійснюється розробка системи. По-перше, калькулятор орієнтований на демонстрацію принципів розрахунку вартості доставки та не претендує на повну відповідність тарифам конкретного оператора — фактичні тарифи можуть бути легко змінені адміністратором системи. По-друге, система не передбачає інтеграції з базами даних населених пунктів та відділень — вибір географії реалізується через спрощений механізм вибору тарифної зони. По-третє, функціонал обмежений розрахунком вартості без можливості оформлення замовлення на доставку [31].

Алгоритм розрахунку вартості доставки, який буде реалізовано в системі, базується на стандартних підходах логістичних операторів та включає такі етапи:

1. Розрахунок об'ємної ваги за формулою:

$$V_w = (L \times W \times H) / 4000,$$

де L , W , H — габарити посилки в сантиметрах.

2. Визначення тарифної ваги як максимуму з фактичної та об'ємної ваги:

$$T_w = \max(F_w, V_w).$$

3. Визначення базової вартості на основі тарифної ваги та обраної тарифної зони.

4. Додавання вартості додаткових послуг (страхування, накладений платіж, кур'єрський збір).

5. Формування підсумкової вартості та її візуалізація користувачеві.

Вибір технологій реалізації обумовлений вимогами до автономності та компактності системи. HTML5 забезпечує семантичну структуру документа та підтримку сучасних форм введення даних (числові поля з валідацією, повзунки). CSS3 дозволяє створити адаптивний дизайн за допомогою медіа-запитів та flexbox/grid-розмітки. JavaScript виконує роль «двигуна» калькулятора, обробляючи події введення та оновлюючи DOM-дерево сторінки в режимі реального часу [32].

Використання зовнішніх бібліотек та фреймворків (React, Vue, jQuery) визнано недоцільним для даного проєкту з кількох причин. По-перше, функціональність калькулятора є достатньо простою, щоб бути реалізованою на «чистому» JavaScript без додаткових абстракцій. По-друге, відмова від зовнішніх залежностей дозволяє досягти мінімального розміру файлу та гарантувати його роботу без підключення до CDN. По-третє, навчальна мета дипломної роботи передбачає демонстрацію базових принципів веб-розробки, а не використання готових рішень [33].

Додатковою вимогою до системи є визначення апаратного забезпечення, необхідного для її функціонування. Ця вимога охоплює як серверну частину (веб-сервер для хостингу HTML-файлу), так і клієнтську частину (пристрої кінцевих користувачів). Детальний аналіз апаратних вимог буде проведено у відповідному розділі дипломної роботи з урахуванням специфіки односторінкового застосунку.

Таким чином, у першому розділі дипломної роботи проведено комплексний аналіз предметної області. Досліджено стан та тенденції розвитку українського ринку послуг доставки, визначено ключових гравців та фактори, що впливають на ціноутворення. Здійснено огляд існуючих програмних рішень калькуляторів вартості доставки, виявлено їх функціональні можливості, переваги та недоліки. На

основі проведеного аналізу сформульовано постановку задачі, визначено функціональні та нефункціональні вимоги до системи, обґрунтовано вибір технологій реалізації.

1.4 Висновки до розділу 1

Український ринок послуг доставки демонструє стабільне зростання, обумовлене розвитком електронної комерції. Обсяг відправлень провідних операторів у 2024 році досяг рекордних показників: «Нова пошта» обробила 480 млн посилок, що підтверджує високий попит на логістичні послуги.

Вартість послуг доставки формується під впливом множини факторів: фізичних характеристик відправлення (вага, габарити), географії доставки (тарифні зони), типу доставки (відділення, кур'єр, поштомат), додаткових послуг (страхування, накладений платіж). Розуміння цих факторів є ключовим для проектування калькулятора.

Наявні калькулятори вартості доставки переважно реалізовані як клієнт-серверні застосунки, що залежать від API логістичних операторів. Це обмежує можливості кастомізації та створює залежність від зовнішніх сервісів.

Архітектура односторінкового застосунку (SPA) є оптимальним рішенням для реалізації калькулятора, оскільки забезпечує миттєву реакцію інтерфейсу, автономність роботи та простоту розгортання.

Технологічний стек HTML5 + CSS3 + JavaScript є достатнім для реалізації поставленої задачі без використання зовнішніх бібліотек та фреймворків, що забезпечує компактність рішення та незалежність від сторонніх ресурсів.

2 ПРОЄКТУВАННЯ ВЕБ-СИСТЕМИ

2.1 Вибір технологій розробки

2.1.1 Обґрунтування вибору HTML5, CSS3, JavaScript

Розробка сучасної веб-системи калькулятора вартості послуг доставки вимагає ретельного вибору технологічного стеку, який забезпечить оптимальне поєднання продуктивності, підтримки браузерами та простоти розгортання. Враховуючи вимоги до проєкту, зокрема реалізацію системи у вигляді односторінкового застосунку без серверної частини, було обрано класичний стек веб-технологій: HTML5, CSS3 та JavaScript.

HTML5 (HyperText Markup Language версії 5) є сучасним стандартом розмітки веб-документів, який було офіційно затверджено консорціумом W3C у 2014 році. На відміну від попередніх версій, HTML5 запроваджує семантичні елементи, що значно покращують структуру документа та його доступність. Згідно з документацією MDN Web Docs, семантичні елементи чітко описують своє призначення як для браузера, так і для розробника, що суттєво відрізняє їх від несемантичних елементів на кшталт `<div>` та `<span>` [1].

До ключових семантичних елементів HTML5 належать: `<header>` – визначає заголовкову частину документа або секції; `<nav>` – містить навігаційні посилання; `<main>` – позначає основний унікальний контент сторінки; `<section>` – групує тематично пов'язаний контент; `<article>` – представляє самодостатній контент; `<footer>` – визначає підвал документа або секції [2]. Використання цих елементів не лише покращує читабельність коду, але й сприяє кращій індексації пошуковими системами та підвищує доступність для асистивних технологій.

HTML5 також запроваджує нові типи полів введення, що є критично важливими для калькулятора доставки. Тип `number` забезпечує введення лише числових значень із можливістю налаштування мінімальних, максимальних значень та кроку. Тип `tel` призначений для телефонних номерів, а тип `email` автоматично валідує формат електронної адреси. Атрибути `required`, `pattern` та

min/max дозволяють реалізувати вбудовану валідацію форм без використання JavaScript, що підвищує надійність системи [3].

CSS3 (Cascading Style Sheets Level 3) є сучасним стандартом каскадних таблиць стилів, що значно розширює можливості стилізації веб-сторінок порівняно з попередніми версіями. Ключовою особливістю CSS3 є модульна архітектура, де різні функціональні можливості розділені на окремі модулі: Flexbox, Grid, Transitions, Animations, Media Queries тощо [4].

Модуль Flexbox (Flexible Box Layout) забезпечує гнучке розташування елементів у одновимірному просторі. Він дозволяє автоматично розподіляти простір між елементами контейнера та вирівнювати їх за різними осями. Flexbox особливо ефективний для створення навігаційних меню, карток товарів та форм введення даних [5]. Для калькулятора доставки Flexbox буде використано для організації елементів форми введення та відображення результатів розрахунку.

Технологія CSS Grid Layout призначена для створення двовимірних макетів і дозволяє одночасно контролювати розташування елементів як по горизонталі, так і по вертикалі. Grid є потужним інструментом для побудови складних інтерфейсів із чітко визначеною структурою. Поєднання Flexbox та Grid дозволяє створювати адаптивні макети, що коректно відображаються на пристроях із різною роздільною здатністю екрана [6].

JavaScript є мовою програмування, яка забезпечує інтерактивність веб-сторінок. Згідно з даними дослідження Stack Overflow Developer Survey 2024, JavaScript залишається найпопулярнішою мовою програмування серед веб-розробників упродовж останніх років [7]. Мова дозволяє маніпулювати Document Object Model (DOM), обробляти події користувача, виконувати обчислення та валідацію даних без звернення до сервера.

Для розроблюваної системи було обрано стандарт ECMAScript 2020 (ES11), який підтримується всіма сучасними браузерами. Цей стандарт включає такі можливості як оператори optional chaining (?.) та nullish coalescing (??), BigInt для роботи з великими числами, Promise.allSettled() для обробки множинних

асинхронних операцій та динамічний імпорт модулів [8]. Ці можливості дозволяють писати більш лаконічний та надійний код.

Вибір саме цього технологічного стеку обумовлений кількома факторами. По-перше, HTML5, CSS3 та JavaScript є відкритими стандартами, що не потребують ліцензійних відрахувань та підтримуються всіма сучасними браузерами. По-друге, клієнтська архітектура дозволяє розгорнути систему на будь-якому статичному хостингу без потреби в серверній інфраструктурі. По-третє, низький поріг входження полегшує подальшу підтримку та модифікацію системи.

Таблиця 2.1 — Порівняння підходів до розробки веб-застосунків

Критерій	HTML/CSS/JS	React/Vue	Angular
Складність впровадження	Низька	Середня	Висока
Розмір бандлу	< 50 КБ	100-300 КБ	500+ КБ
Залежність від збірки	Відсутня	Необхідна	Необхідна
Підтримка браузерами	Нативна	Через транспіляцію	Через транспіляцію
Придатність для малих проєктів	Оптимальна	Надлишкова	Надлишкова

Як видно з табл. 2.1, для розробки невеликого односторінкового застосунку використання класичного стеку HTML/CSS/JS є найбільш раціональним рішенням. Фреймворки React, Vue чи Angular доцільно застосовувати для масштабних проєктів із складною бізнес-логікою та необхідністю повторного використання компонентів.

2.1.2 Архітектура односторінкового застосунку (SPA)

Односторінковий застосунок (Single Page Application, SPA) — це веб-застосунок, який взаємодіє з користувачем шляхом динамічного перезапису

поточної сторінки замість завантаження повністю нових сторінок із сервера. Такий підхід забезпечує швидші переходи та створює враження роботи з нативним застосунком [9]. Концепція SPA почала активно розвиватися з 2010-х років разом із появою потужних JavaScript-фреймворків.

Архітектура SPA принципово відрізняється від традиційних багатосторінкових застосунків (Multi-Page Application, MPA). У класичній моделі MPA кожна дія користувача призводить до запиту на сервер та повного перезавантаження сторінки. У SPA після початкового завантаження HTML, CSS та JavaScript-коду вся подальша взаємодія відбувається на стороні клієнта. Сервер, якщо він використовується, виконує роль API для обміну даними [10].

Основні переваги архітектури SPA для розроблюваної системи включають: покращений користувацький досвід завдяки відсутності перезавантажень сторінки; зменшене навантаження на сервер, оскільки обчислення виконуються на клієнті; спрощене розгортання — достатньо статичного хостингу; можливість офлайн-роботи за умови кешування ресурсів [11].

Водночас SPA має певні обмеження. Пошукова оптимізація (SEO) є складнішою, оскільки пошукові роботи традиційно погано індексують JavaScript-контент. Однак для калькулятора доставки, який є інструментальним застосунком, а не контентним сайтом, це обмеження не є критичним. Google офіційно оголосив про здатність індексувати JavaScript-контент, хоча рекомендує серверний рендеринг для контентних сайтів [12].

Для реалізації SPA без використання фреймворків застосовується патерн проєктування Module Pattern, який дозволяє інкапсулювати логіку та запобігати забрудненню глобального простору імен. Кожен функціональний модуль системи (валідація, розрахунок, відображення) оформлюється як самодостатній JavaScript-модуль із чітко визначеним публічним інтерфейсом [13].

Взаємодія з DOM (Document Object Model) у SPA оптимізується через мінімізацію прямих маніпуляцій. Замість частих звернень до DOM, дані спочатку обробляються в пам'яті, а потім результати відображаються за один прохід. Це

значно підвищує продуктивність застосунку, особливо на мобільних пристроях із обмеженими ресурсами [14].

Керування станом у простих SPA може здійснюватися без спеціалізованих бібліотек. Для калькулятора доставки достатньо зберігати поточні значення полів форми та результати розрахунку в локальних змінних JavaScript. При необхідності збереження даних між сесіями можна використовувати Web Storage API (localStorage або sessionStorage), що забезпечує збереження до 5-10 МБ даних у браузері користувача [15].

2.2 Проєктування інтерфейсу користувача

2.2.1 Макети екранних форм

Проєктування інтерфейсу користувача (User Interface, UI) є важливим етапом розробки, який безпосередньо впливає на зручність використання системи та задоволеність користувачів. Для калькулятора вартості доставки було спроектовано інтуїтивно зрозумілий інтерфейс, що відповідає сучасним стандартам веб-дизайну та забезпечує ефективне виконання основних сценаріїв використання.

Структура інтерфейсу калькулятора базується на принципі поетапного введення даних. Головна екранна форма поділена на логічні блоки: параметри відправлення (вага, габарити), адресна інформація (звідки — куди), тип доставки та додаткові послуги. Такий поділ дозволяє користувачеві послідовно заповнювати форму, не відчуваючи когнітивного перевантаження [16]. Візуальна ієрархія інтерфейсу побудована за принципом F-патерну сканування, який є природним для західної культури читання. Найважливіші елементи — поля введення ваги та габаритів — розташовані у верхній лівій частині форми. Кнопка розрахунку винесена в окремий візуально виділений блок, що привертає увагу користувача після заповнення всіх полів [17].

Макет передбачає три основні стани інтерфейсу: початковий стан з порожніми полями та неактивною кнопкою розрахунку; стан заповнення з валідацією введених даних у реальному часі; стан результату з відображенням розрахованої вартості та деталізації. Переходи між станами супроводжуються візуальним зворотним зв'язком для підтвердження дій користувача.

Для відображення результатів розрахунку передбачено спеціальний блок, який з'являється після натискання кнопки «Розрахувати». Блок містить загальну вартість доставки, виділену великим шрифтом, та деталізовану інформацію про складові вартості: базовий тариф, надбавку за вагу, надбавку за габарити, вартість додаткових послуг. Така структура забезпечує прозорість ціноутворення та підвищує довіру користувачів.

Макет також передбачає адаптивність до різних розмірів екрана. На настільних комп'ютерах форма відображається в два стовпці для оптимального використання простору. На планшетах та смартфонах елементи перебудовуються в один стовпець для зручності взаємодії пальцями. Мінімальний розмір елементів керування встановлено 44×44 пікселі відповідно до рекомендацій Apple Human Interface Guidelines [18].

2.2.2 UX/UI принципи побудови інтерфейсу

Користувацький досвід (User Experience, UX) охоплює всі аспекти взаємодії користувача з системою. Якісний UX-дизайн забезпечує інтуїтивність, ефективність та задоволення від використання продукту. При проектуванні калькулятора доставки було застосовано ключові принципи UX-дизайну, що підтверджені багаторічними дослідженнями в галузі людино-комп'ютерної взаємодії.

Принцип простоти (Simplicity) передбачає мінімізацію когнітивного навантаження на користувача. Інтерфейс калькулятора містить лише необхідні елементи без візуального шуму. Кожне поле має чітку мітку та, за потреби, підказку. Згідно з законом Хіка, час прийняття рішення зростає логарифмічно зі збільшенням кількості варіантів, тому кількість опцій у випадальних списках обмежено найбільш релевантними значеннями [19].

Принцип послідовності (Consistency) забезпечує передбачуваність поведінки інтерфейсу. Всі елементи керування виконані в єдиному візуальному стилі: однакові відступи, шрифти, кольори для однотипних елементів. Користувач, навчившись взаємодіяти з одним елементом, автоматично розуміє, як працювати з іншими [20].

Принцип зворотного зв'язку (Feedback) вимагає негайного реагування системи на дії користувача. При введенні даних у поля форми система миттєво валідує значення та відображає статус: зелена галочка для коректних даних, червоний індикатор з описом помилки для некоректних. Під час розрахунку відображається індикатор завантаження, навіть якщо обчислення виконуються миттєво — це підтверджує, що система отримала команду [21].

Принцип доступності (Accessibility) гарантує можливість використання системи людьми з різними можливостями. Калькулятор розроблено з урахуванням рекомендацій WCAG 2.1 (Web Content Accessibility Guidelines): достатній контраст тексту (мінімум 4.5:1), можливість навігації клавіатурою, коректна робота з програмами читання екрана завдяки семантичній розмітці та ARIA-атрибутам [22].

Закон Фіттса визначає, що час досягнення цілі залежить від відстані до неї та її розміру. Відповідно, кнопка «Розрахувати» має збільшений розмір та розташована поблизу основних полів введення. На мобільних пристроях кнопка займає всю ширину екрана для максимальної зручності натискання [23].

Закон близькості (Gestalt Proximity) використано для візуального групування пов'язаних елементів. Поля ваги та габаритів розташовані поруч у блоці «Параметри відправлення». Адреси відправника та отримувача об'єднані в блок «Маршрут». Візуальна близькість допомагає користувачеві швидше орієнтуватися в інтерфейсі [24].

Колірна схема інтерфейсу базується на принципах теорії кольору та психології сприйняття. Основним кольором обрано синій (#2563EB), який асоціюється з надійністю та професійністю. Акцентний зелений (#10B981) використовується для підтвердження успішних дій. Червоний (#EF4444) сигналізує

про помилки. Нейтральні сірі відтінки забезпечують візуальний баланс та не відволікають від основного контенту [25].

2.3 Проєктування логіки розрахунку вартості

2.3.1 Алгоритм калькуляції

Алгоритм розрахунку вартості доставки є ключовим компонентом системи, який визначає точність та корисність калькулятора. На основі аналізу тарифних політик провідних операторів доставки в Україні було розроблено узагальнений алгоритм, що враховує основні фактори ціноутворення: фізичну вагу, об'ємну (габаритну) вагу, відстань доставки та тип послуги.

Вартість доставки розраховується за формулою:

$$C = B + (W \times R_w) + (D \times R_d) + S,$$

де C — загальна вартість доставки;

B — базовий тариф;

W — розрахункова вага (максимум з фізичної та об'ємної);

R_w — ставка за кілограм ваги;

D — відстань у кілометрах;

R_d — ставка за кілометр;

S — сумарна вартість додаткових послуг [26].

Об'ємна (габаритна) вага розраховується за стандартною формулою, що використовується в логістичній галузі:

$$W_{vol} = (L \times W \times H) / K,$$

де L , W , H — довжина, ширина та висота відправлення в сантиметрах;

K — коефіцієнт об'ємної ваги.

Типове значення K для автомобільних перевезень становить 4000-5000, для авіаперевезень — 6000. У калькуляторі використовується значення 4000, що відповідає практиці більшості українських операторів [27].

Алгоритм визначення розрахункової ваги порівнює фізичну та об'ємну вагу і обирає більшу з них. Цей підхід, відомий як «dimensional weight pricing», дозволяє операторам доставки коректно враховувати вартість перевезення легких, але об'ємних вантажів, які займають значний простір у транспортному засобі [28].

Таблиця 2.2 — Базові тарифи за типами доставки

Тип доставки	Базовий тариф	Ставка за кг	Ставка за км
Стандартна (склад-склад)	35 грн	5 грн	0.10 грн
Адресна (до дверей)	55 грн	7 грн	0.15 грн
Експрес (прискорена)	75 грн	10 грн	0.20 грн
У межах міста	45 грн	6 грн	—

Тарифи, наведені в табл. 2.2, є демонстраційними та базуються на аналізі цінових пропозицій провідних операторів ринку. У виробничій системі тарифи мають завантажуватися з конфігураційного файлу або отримуватися через API для забезпечення актуальності.

2.3.2 Діаграма послідовності обчислень

Послідовність обчислень у калькуляторі доставки представлена у вигляді алгоритму, що складається з декількох етапів. На першому етапі система отримує вхідні дані від користувача: фізичну вагу відправлення, габаритні розміри (довжина, ширина, висота), адреси відправника та отримувача, тип доставки та перелік обраних додаткових послуг.

Другий етап передбачає валідацію введених даних. Перевіряється коректність числових значень (вага та габарити мають бути додатними числами), наявність обов'язкових полів та допустимість обраних опцій. У разі виявлення

помилку процес переривається, користувачеві відображається повідомлення з описом проблеми.

На третьому етапі відбувається розрахунок проміжних значень. Обчислюється об'ємна вага за формулою габаритної ваги. Визначається розрахункова вага як максимум з фізичної та об'ємної. Розраховується відстань між адресами за допомогою таблиці відстаней або геокодування (у спрощеній версії — за номером зони доставки).

Четвертий етап — безпосередній розрахунок вартості. На основі типу доставки з таблиці тарифів обираються відповідні ставки. Обчислюється базова вартість як сума базового тарифу, надбавки за вагу та надбавки за відстань. Додається вартість обраних додаткових послуг: страхування, упакування, зворотної доставки документів тощо.

П'ятий етап передбачає форматування результату для відображення користувачеві. Загальна вартість округлюється до копійок. Формується деталізований звіт із зазначенням кожної складової вартості. Результат передається до модуля відображення, який оновлює відповідні елементи DOM.

Лістинг 2.1 - Псевдокод основної функції розрахунку

```

ФУНКЦІЯ розрахуватиВартість(дані) :
ЯКЩО НЕ валідувати(дані) ТОДІ
    ПОВЕРНУТИ помилка
обємнаВага = (дані.довжина × дані.ширина × дані.висота) / 4000
розрахВага = МАКСИМУМ(дані.вага, обємнаВага)
тариф = отриматиТариф(дані.типДоставки)
відстань = визначитиВідстань(дані.звідки, дані.куди)
вартість = тариф.база + розрахВага × тариф.заКг
+ відстань × тариф.заКм
ДЛЯ КОЖНОЇ послуги В дані.додатковіПослуги:
    вартість = вартість + послуга.ціна
ПОВЕРНУТИ округлити(вартість, 2)

```

2.3.3 Структура даних (тарифи, зони доставки)

Для зберігання тарифів та конфігураційних даних калькулятора обрано формат JSON (JavaScript Object Notation), який є нативним для JavaScript та забезпечує зручне читання як машиною, так і людиною. Структура даних спроектована з урахуванням можливості розширення та легкого оновлення тарифів без модифікації програмного коду.

Основний конфігураційний об'єкт містить три ключові секції: типи доставки з відповідними тарифами, зони доставки з коефіцієнтами та перелік додаткових послуг. Типи доставки описуються об'єктами, що містять назву для відображення, базовий тариф, ставки за вагу та відстань, а також орієнтовний термін доставки.

Зони доставки використовуються для спрощеного розрахунку відстані. Територія України розділена на зони за регіональним принципом: внутрішньоміська доставка (зона 1), доставка в межах області (зона 2), доставка в сусідні області (зона 3) та міжрегіональна доставка (зона 4). Кожна зона має фіксований коефіцієнт, що множиться на базову ставку за відстань [29].

Додаткові послуги представлені масивом об'єктів, де кожен об'єкт містить унікальний ідентифікатор, назву, опис та вартість. Вартість може бути фіксованою або відсотковою від оголошеної вартості відправлення. Для страхування застосовується відсоткова ставка, для упакування та інших послуг — фіксована сума.

Валідація вхідних даних забезпечується за допомогою схем валідації. Для кожного поля введення визначено тип даних, діапазон допустимих значень та повідомлення про помилку. Такий підхід дозволяє централізовано керувати правилами валідації та легко локалізувати повідомлення для різних мов [30].

2.4 Структура HTML-документа

2.4.1 Семантична розмітка

Семантична розмітка HTML5 передбачає використання елементів, назви яких відображають їхнє призначення та роль у структурі документа. На відміну від універсальних контейнерів `<div>`, семантичні елементи надають додаткову інформацію браузерам, пошуковим системам та асистивним технологіям про зміст та структуру сторінки [31].

Структура HTML-документа калькулятора доставки організована за принципом логічного поділу на функціональні блоки. Кореневий елемент `<html>` містить атрибут `lang="uk"` для коректної обробки українського тексту. Секція `<head>` включає метадані документа, налаштування `viewport` для адаптивності та підключення стилів.

Основна структура тіла документа (`<body>`) складається з трьох семантичних блоків: `<header>` — заголовок сторінки з назвою калькулятора та логотипом; `<main>` — основний контент із формою введення та блоком результатів; `<footer>` — підвал з контактною інформацією та копірайтом. Така структура відповідає рекомендаціям W3C та забезпечує доступність для програм читання екрана [32].

Елемент `<main>` містить єдиний унікальний контент сторінки. Всередині `<main>` розміщено елемент `<form>` для введення даних калькулятора. Форма логічно поділена на секції за допомогою елементів `<fieldset>`, кожна з яких групує пов'язані поля введення. Елемент `<legend>` надає заголовок для кожної групи, що покращує доступність.

Для полів введення використовуються сучасні типи `input` HTML5. Поле ваги має тип `number` з атрибутами `min`, `max` та `step` для обмеження введення. Поля габаритів також мають тип `number`. Випадаючі списки міст реалізовані через `<select>` з елементами `<optgroup>` для групування за регіонами. Перемикачі типу доставки використовують `<input type="radio">`, а чекбокси додаткових послуг — `<input type="checkbox">`.

Кожне поле введення асоціюється з міткою `<label>` через атрибут `for`, що забезпечує розширену область натискання та коректну роботу з програмами читання екрана. Підказки та повідомлення про помилки розміщуються в елементах `<span>` з відповідними ARIA-атрибутами для забезпечення доступності [33].

Блок результатів розрахунку використовує елемент `<output>` для відображення обчисленої вартості. Цей семантичний елемент спеціально призначений для відображення результатів обчислень або дій користувача. Деталізація вартості представлена у вигляді списку визначень `<dl>` з елементами `<dt>` для назв складових та `<dd>` для їх значень.

2.4.2 Модульна організація коду

Модульна організація коду є фундаментальним принципом програмної інженерії, що забезпечує розділення відповідальності, повторне використання компонентів та спрощення тестування. Для односторінкового застосунку, реалізованого в єдиному HTML-файлі, модульність досягається через структурування JavaScript-коду у вигляді незалежних функціональних модулів.

Код калькулятора організовано за патерном `Revealing Module Pattern`, який дозволяє інкапсулювати приватні змінні та функції, відкриваючи лише необхідний публічний інтерфейс. Кожен модуль оформлюється як самовиконувана функція (IIFE — `Immediately Invoked Function Expression`), що повертає об'єкт з публічними методами [34].

Система поділена на наступні логічні модулі: `ConfigModule` – зберігає тарифи та налаштування; `ValidationModule` – відповідає за валідацію вхідних даних; `CalculationModule` – виконує розрахунок вартості; `UIModule` – керує відображенням та взаємодією з DOM; `AppModule` – координує роботу всіх модулів та ініціалізує застосунок.

CSS-код організовано за методологією BEM (`Block Element Modifier`), що забезпечує передбачувані назви класів та уникає конфліктів стилів. Блоки відповідають логічним компонентам інтерфейсу: `.calculator`, `.form-group`, `.result-block`. Елементи позначаються подвійним підкресленням: `.calculator__title`, `.form-`

group__label. Модифікатори вказують на стан елемента: .form-group--error, .button--primary [35].

Для підвищення продуктивності CSS-стилі розміщуються у секції <head> у вигляді вбудованих стилів (<style>), а JavaScript-код – перед закриваючим тегом </body>. Такий порядок забезпечує рендеринг сторінки до завантаження скриптів, що покращує показник First Contentful Paint (FCP) та загальний користувацький досвід [36].

Альтернативою вбудованим стилям та скриптам є використання Base64-кодування для вбудовування зовнішніх ресурсів безпосередньо в HTML через Data URI. Однак для калькулятора доставки, де загальний обсяг коду не перевищує декількох десятків кілобайт, вбудований підхід є оптимальним з точки зору кількості HTTP-запитів та простоти розгортання.

2.5 Висновки до розділу 2

У другому розділі дипломної роботи виконано проєктування веб-системи калькулятора вартості послуг доставки. Обґрунтовано вибір технологічного стеку HTML5, CSS3 та JavaScript як оптимального рішення для розробки односторінкового застосунку без серверної частини. Проведено порівняльний аналіз з альтернативними підходами на базі фреймворків React, Vue та Angular.

Розглянуто архітектуру односторінкового застосунку (SPA), її переваги та обмеження. Описано принципи клієнтського рендерингу, керування станом та взаємодії з DOM. Обґрунтовано застосовність архітектури SPA для калькулятора доставки з урахуванням вимог до продуктивності та простоти розгортання.

Спроектовано інтерфейс користувача з урахуванням принципів UX/UI дизайну. Розроблено макети екранних форм для різних станів системи. Застосовано ключові закони UX-дизайну: закон Хіка, закон Фіттса, закон близькості Гештальт-теорії. Обрано колірну схему, що відповідає принципам теорії кольору та забезпечує доступність.

Розроблено алгоритм розрахунку вартості доставки, що враховує фізичну та об'ємну вагу, відстань, тип доставки та додаткові послуги. Визначено структуру даних для зберігання тарифів та конфігурації системи у форматі JSON.

Описано структуру HTML-документа з використанням семантичної розмітки HTML5. Застосовано модульну організацію коду за патерном Revealing Module Pattern та методологію BEM для структурування CSS-стилів. Визначено принципи оптимізації продуктивності односторінкового застосунку.

3 РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ

3.1 Розробка інтерфейсної частини

Інтерфейс калькулятора вартості доставки побудовано відповідно до принципів юзабіліті та доступності, визначених у підрозділі 2.2. Структура HTML-документа базується на семантичних елементах HTML5, що забезпечує логічну організацію вмісту та покращену доступність для асистивних технологій [3]. Головна сторінка системи включає три ключові семантичні блоки: заголовок (header), основний контент (main) та підвал (footer), кожен з яких виконує чітко визначену функцію в загальній архітектурі інтерфейсу.

Заголовок сайту реалізовано як фіксований елемент (sticky header) із застосуванням властивості CSS `position: sticky`, що забезпечує його постійну видимість під час прокручування сторінки. Для досягнення ефекту «розмитого скла» (glassmorphism) використано властивість `backdrop-filter: blur(20px)`, яка створює напівпрозорий фон із розмиттям контенту, що знаходиться позаду заголовка [4]. Такий підхід відповідає сучасним тенденціям у дизайні інтерфейсів та забезпечує візуальне відокремлення навігаційної зони від основного вмісту.

Логотип системи «DeliveryCost Pro» реалізовано із використанням CSS-градієнта та властивості `background-clip: text`, що дозволяє застосувати градієнтну заливку безпосередньо до тексту. Поруч із логотипом розміщено індикатор статусу

«Онлайн-розрахунок» з анімованою пульсуючою точкою, реалізованою через CSS-анімацію `@keyframes pulse`, що візуально підтверджує працездатність сервісу.

Основний контент сторінки організовано за допомогою CSS Grid Layout із двоколонковою структурою. На десктопних пристроях (ширина екрана понад 1024 пікселів) використовується розподіл `grid-template-columns: 1.2fr 0.8fr`, де ліва колонка із формою введення даних займає більшу частину простору, а права колонка з результатами розрахунку — меншу. На мобільних пристроях макет автоматично перебудовується в одноколонковий режим завдяки медіа-запиту `@media (max-width: 1024px)` [5].

На рис. 3.1 наведено загальний вигляд інтерфейсу калькулятора вартості доставки у браузері. Як видно, інтерфейс складається з двох основних панелей: лівої панелі з формою введення параметрів відправлення та правої панелі з результатами розрахунку і тарифною інформацією.

Форма введення даних організована у три тематичні секції, кожна з яких обрамлена елементом `fieldset` із візуальним заголовком. Перша секція «Параметри вантажу» містить поля для введення фактичної ваги та габаритних розмірів посилки. Друга секція «Параметри доставки» включає вибір тарифної зони, типу доставки та пункту призначення. Третя секція «Додаткові послуги» надає можливість обрати страхування та супутні сервіси. Такий поділ відповідає принципу прогресивного розкриття інформації (*progressive disclosure*), коли користувач проходить логічну послідовність кроків від характеристик вантажу до параметрів доставки та додаткових опцій [6].

Форма введення даних реалізована з використанням семантичних HTML5-елементів, що забезпечують вбудовану браузерну валідацію та покращену взаємодію на мобільних пристроях. Поля для введення числових даних (вага, габарити) використовують тип `input type="number"` з атрибутами `min`, `max` та `step`, що обмежують діапазон допустимих значень та крок зміни. Наприклад, поле ваги має мінімальне значення 0.1 кг, максимальне — 1000 кг, і крок 0.1 кг, що відповідає реальним потребам логістичної практики [7].

Розрахунок вартості доставки
Введіть параметри вашого відправлення та отримаєте миттєвий розрахунок вартості з детальною деталізацією всіх складових.

Параметри відправлення

Не користувалися?
Додавайте нові типи посилок (розмірами*), обирайте тип доставки та заповнюйте результати. Розрахунок з урахуванням витрат на експрес-доставку.

Параметри вантажу

Фізична вага *

Внутрішня вага *

Габарити посилок *

Довжина: [input type="text"] Ширина: [input type="text"] Висота: [input type="text"]

Параметри доставки

Зона доставки *

Область зони *

Тип доставки *

☒ Стандартна 3-5 робочих днів ☐ Бюджетна 7-10 робочих днів ☐ Титовий 1-2 робочих днів

Пункт призначення *

☒ До відправника ☐ До поштової ☐ Адресна доставка

Додаткові послуги

Опційна вартість (для експрес-доставки)

Оплатити додаткові опції

☐ Пакування ☐ Кредитний вантаж ☐ Інші послуги

☐ Інші послуги ☐ Завантаження доставки

Результат розрахунку

Заповнюйте форму
Введіть параметри відправлення у формі зони, щоб побачити розрахунок вартості доставки.

Зона	вагою до 1 кг
Зона 1 (місто)	35 грн 5 грн/кг
Зона 2 (область)	45 грн 7 грн/кг
Зона 3 (районська)	55 грн 9 грн/кг
Зона 4 (сільська)	70 грн 12 грн/кг

Розрахувати вартість

Оформити форму

Рисунок 3.1 — Загальний вигляд інтерфейсу калькулятора вартості доставки

Габаритні розміри посилки (довжина, ширина, висота) розміщено в одному рядку за допомогою CSS Grid з конфігурацією `grid-template-columns: repeat(3, 1fr)`, що забезпечує рівномірний розподіл простору між трьома полями. На мобільних пристроях (ширина екрана менше 640 пікселів) ці поля автоматично перебудовуються у вертикальний стек завдяки медіа-запиту, який змінює конфігурацію сітки на `grid-template-columns: 1fr` [8].

На рисунку 3.2 детально показано форму введення параметрів вантажу та доставки з заповненими даними. Зокрема, видно поля введення ваги (1,3 кг), габаритних розмірів (12×36×26 см), вибір тарифної зони та типу доставки. Вибір типу доставки реалізовано за допомогою інтерактивних радіо-карток (option

cards), що замінюють стандартні елементи radio. Кожна картка містить назву типу доставки, іконку, орієнтовний термін та коефіцієнт множення вартості. Реалізація базується на прихованому елементі `input[type="radio"]` з класом `option-card__input`, який має стиль `opacity: 0` та `position: absolute`, а візуальне відображення забезпечується суміжним елементом `option-card__content`. CSS-селектор `input:checked + .option-card__content` змінює стиль обраної картки, додаючи синю рамку та фонове підсвічування [9]. За замовчуванням встановлено значення «Стандартна» (атрибут `checked`), що усуває можливість відправлення форми без обраного типу доставки.

Параметри відправлення

Як користуватися

Заповніть обов'язкові поля (позначені *), оберіть тип доставки та додаткові послуги. Результат з'явиться миттєво після натискання кнопки «Розрахувати».

Параметри вантажу

Фактична вага *

1,3

кг · Від 0.1 до 1000 кг

Габарити посилок *

12

CM

36

CM

26

CM

Параметри доставки

Зона доставки *

Зона 1: Місто (до 30 км) — 35 грн базовий

Тип доставки *

☐

Стандартна

3-5 робочих днів

×1.0

☒

Експрес

1-2 робочих днів

×1.5

☐

Того ж дня

У межах міста

×2.0

Пункт призначення *

До відділення

Отримання у найближчому відділенні

Безкоштовно

Результат розрахунку

Заповніть форму

Введіть параметри відправлення у формі зліва, щоб побачити розрахунок вартості доставки.

Тарифна сітка

ЗОНА	БАЗОВИЙ	ЗА 1 КГ
Зона 1 (місто)	35 грн	5 грн/кг
Зона 2 (область)	45 грн	7 грн/кг
Зона 3 (міжобласна)	55 грн	9 грн/кг
Зона 4 (віддалена)	70 грн	12 грн/кг

Рисунок 3.2 — Форма введення параметрів вантажу та доставки із заповненими даними

Бейдж «Популярна» на картці експрес-доставки реалізовано через абсолютно позиційований елемент із класом `option-card__badge`, розміщений у верхньому правому куті картки. Це рішення базується на принципі візуальної підказки (visual cue), що привертає увагу користувача до найбільш затребуваної опції та полегшує процес прийняття рішення.

Стилізація калькулятора побудована на системі CSS Custom Properties (змінних), що забезпечує централізоване управління дизайн-токенами та спрощує підтримку візуальної консистентності інтерфейсу. У кореновому елементі `:root` визначено понад 70 змінних, згрупованих за категоріями: брендові кольори (`--brand-primary`, `--brand-secondary`), семантичні кольори (`--success-500`, `--error-500`, `--warning-500`), нейтральна палітра (`--gray-50 ... --gray-900`), типографічна шкала (`--text-xs ... --text-5xl`), система відступів (`--space-1 ... --space-20`), тіні (`--shadow-sm ... --shadow-2xl`) та радіуси заокруглення (`--radius-sm ... --radius-3xl`) [10].

Такий підхід до організації стилів відповідає концепції дизайн-системи і дозволяє змінити візуальну тему всього застосунку, модифікувавши значення змінних в одному місці. Наприклад, зміна значення `--brand-primary` з `#2563EB` на інший колір автоматично оновить усі елементи інтерфейсу, що використовують цю змінну: кнопки, посилання, активні стани карток, індикатори секцій тощо.

Фоновий градієнт сторінки (`linear-gradient(135deg, #667eea 0%, #764ba2 100%)`) створює візуальну глибину та контраст із білими картками основного контенту. Картки (клас `.card`) мають білий фон, заокруглення `border-radius: 1rem` та тінь `box-shadow` рівня `--shadow-xl`, що забезпечує ефект піднесення над фоновою поверхнею (elevation). При наведенні курсора картки анімовано піднімаються на 4 пікселі (`transform: translateY(-4px)`) із збільшенням тіні до рівня `--shadow-2xl`, що створює зворотний зв'язок та покращує інтерактивність [11].

На рис. 3.3 продемонстровано секцію вибору пункту призначення та додаткових послуг. Видно інтерактивні картки для обрання типу доставки (до відділення, до поштомоту, адресна доставка) з відповідними цінами та коефіцієнтами, а також чекбокс-картки додаткових сервісів.

Реалізацію адресної доставки спроектовано із динамічним відображенням додаткових полів. При виборі опції «Адресна доставка» автоматично з'являється блок полів для введення адреси: місто, вулиця, будинок, квартира та коментар для кур'єра. Блок адреси (`id="addressFields"`) за замовчуванням прихований через стиль `display: none`, а при активації радіо-кнопки «Адресна доставка» JavaScript-обробник додає CSS-клас `visible`, що змінює стиль на `display: block` [12]. Анімація появи реалізована через CSS `@keyframes slideDown`, яка плавно зсуває та проявляє елемент протягом 300 мілісекунд.

Стандартна ×1.0
3-5 робочих днів

Експрес ×1.5
1-2 робочих дні

Того ж дня ×2.0
У межах міста

Пункт призначення *

До відділення Безкоштовно
Отримання у найближчому відділенні

До поштомату -5%
Цілодобовий доступ, знижка -5%

Адресна доставка +15%
Кур'єр доставить за вказаною адресою

Додаткові послуги

Оголошена вартість (для страхування)
0
грн · Комісія: 0.5% від оголошеної вартості

Оберіть додаткові опції

Пакування +20 грн
Захисне пакування вантажу

Крихкий вантаж +35 грн
Додатковий захист та маркування

SMS-сповіщення +5 грн
Повідомлення про статус доставки

Зворотна доставка +30 грн
Повернення підписаних документів

Розрахувати вартість

Очистити форму

Рисунок 3.3 — Секція вибору пункту призначення та додаткових послуг

Валідація адресних полів інтегрована у загальний процес перевірки форми: якщо обрано адресну доставку, система перевіряє обов'язковість полів «Місто», «Вулиця» та «Будинок», інформуючи користувача про незаповнені поля. Поле «Квартира/Офіс» та «Коментар для кур'єра» є необов'язковими, що позначено відповідною міткою «(необов'язково)».

Лістинг 3.1 — CSS-стилі для анімації блоку адресних полів

```
.address-fields {
  display: none;
  margin-top: var(--space-4);
  padding-top: var(--space-4);
  border-top: 1px dashed var(--gray-300);
  animation: slideDown 0.3s ease-out;
}

.address-fields.visible {
  display: block;
}

@keyframes slideDown {
  from {
    opacity: 0;
    transform: translateY(-10px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
```

Кнопка «Розрахувати вартість» є основним елементом взаємодії та стилізована із застосуванням градієнтного фону (linear-gradient від --brand-primary до --brand-primary-dark) та об'ємної тіні синього відтінку (box-shadow: 0 4px 14px rgba(37, 99, 235, 0.39)). При натисканні кнопка демонструє стан завантаження: з'являється CSS-спінер (анімований border-radius: 50% елемент із неповною рамкою), а текст кнопки стає напівпрозорим. Це забезпечує зворотний зв'язок, підтверджуючи, що система обробляє запит.

Типографіка інтерфейсу базується на шрифті Inter (основний шрифт інтерфейсу) та JetBrains Mono (моноширинний шрифт для відображення числових

значень у результатах). Шрифти підключено через Google Fonts API із директивою `font-display: swap`, яка забезпечує миттєве відображення тексту системним шрифтом із подальшою заміною на завантажений веб-шрифт [13]. Антиаліасинг тексту оптимізовано через властивості `-webkit-font-smoothing: antialiased` та `-moz-osx-font-smoothing: grayscale`, що забезпечує чіткість шрифтів на екранах різної роздільної здатності.

3.2 Реалізація бізнес-логіки

Бізнес-логіка калькулятора реалізована у вигляді двох JavaScript-модулів: `Calculator` (обчислювальне ядро) та `UI` (контролер інтерфейсу). Кожен модуль побудований за патерном `Revealing Module Pattern`, що реалізується через самовикликану функцію (IIFE — `Immediately Invoked Function Expression`), яка повертає об'єкт із публічним API, зберігаючи приватність внутрішніх змінних та методів [14].

Модуль `Calculator` інкапсулює всю обчислювальну логіку системи та надає два публічних методи: `calculate(data)` для виконання розрахунку і `validate(data)` для перевірки вхідних даних. Усі тарифні дані зберігаються у приватних константах модуля, що унеможливорює їх модифікацію ззовні [15].

Тарифна конфігурація представлена трьома об'єктами-словниками. Перший — `ZONE_TARIFFS` — містить тарифні параметри для чотирьох зон доставки: базовий тариф (`baseFee`) та ставку за кілограм ваги (`perKg`). Другий — `DELIVERY_MULTIPLIERS` — визначає коефіцієнти для трьох типів доставки: стандартної (1.0), експресної (1.5) та доставки того ж дня (2.0). Третій — `DESTINATION_MULTIPLIERS` — задає коефіцієнти залежно від пункту призначення: до відділення (1.0), до поштомату (0.95 — знижка 5%) та адресна доставка (1.15 — надбавка 15%).

Лістинг 3.2 — Конфігурація тарифних констант модуля Calculator

```

const VOLUMETRIC_DIVISOR = 4000;
const DECLARED_VALUE_RATE = 0.005; // 0.5%

const ZONE_TARIFFS = {
  '1': { baseFee: 35, perKg: 5, name: 'Місто' },
  '2': { baseFee: 45, perKg: 7, name: 'Область' },
  '3': { baseFee: 55, perKg: 9, name: 'Міжобласна' },
  '4': { baseFee: 70, perKg: 12, name: 'Віддалена' }
};

const DELIVERY_MULTIPLIERS = {
  'standard': 1.0,
  'express': 1.5,
  'sameday': 2.0
};

const DESTINATION_MULTIPLIERS = {
  'branch': 1.0,
  'poshtamat': 0.95,
  'address': 1.15
};

```

Алгоритм розрахунку вартості доставки реалізований у функції `calculate(data)`, яка приймає об'єкт із параметрами відправлення та повертає об'єкт із деталізованим результатом. Послідовність обчислень відповідає алгоритму, описаному в підрозділі 2.3, та включає такі етапи:

1. Розрахунок об'ємної ваги за формулою: $V_w = (L \times W \times H) / 4000$, де L , W , H — габаритні розміри посилки у сантиметрах. Дільник 4000 є стандартним коефіцієнтом, що використовується більшістю логістичних операторів для перерахунку об'єму у вагу [16].

2. Визначення розрахункової (тарифної) ваги як максимуму з фактичної та об'ємної ваги: $T_w = \max(F_w, V_w)$. Це гарантує, що вартість доставки враховує як фізичну вагу, так і об'ємність вантажу, відповідно до практики галузі.

3. Обчислення базової вартості доставки за формулою: $C = B + (T_w \times R_w)$, де B — базовий тариф зони, T_w — розрахункова вага, R_w — ставка за кілограм у відповідній тарифній зоні.

4. Застосування множників: вартість множиться на коефіцієнт типу доставки та коефіцієнт пункту призначення.

5. Додавання комісії за оголошену вартість (0.5% від суми) та фіксованих вартостей обраних додаткових послуг.

6. Округлення підсумкової суми до двох знаків після коми функцією `round()`, що реалізує математичне округлення: `Math.round(num × 100) / 100`.

Лістинг 3.3 — Реалізація основної функції розрахунку вартості

```
function calculate(data) {
  // Об'ємна вага
  const volumetricWeight = (data.length * data.width *
    data.height) / VOLUMETRIC_DIVISOR;

  // Розрахункова вага (max з фактичної та об'ємної)
  const chargeableWeight = Math.max(
    data.weight, volumetricWeight
  );

  // Тариф зони
  const zoneTariff = ZONE_TARIFFS[data.zone];

  // Базова вартість = базовий + (вага * ставка)
  const baseCost = zoneTariff.baseFee +
    (chargeableWeight * zoneTariff.perKg);

  // Множники типу та призначення
  const deliveryMult = DELIVERY_MULTIPLIERS[data.deliveryType];
  const destMult = DESTINATION_MULTIPLIERS[data.destinationType];

  // Вартість з коефіцієнтами
  const costWithMult = baseCost * deliveryMult * destMult;

  // Комісія за оголошену вартість та додаткові послуги
  const declaredFee = (data.declaredValue || 0) *
    DECLARED_VALUE_RATE;
  let servicesCost = 0;
  for (const key in data.services) {
    if (data.services[key]) servicesCost +=
      data.services[key];
  }

  return {
    chargeableWeight: round(chargeableWeight),
    total: round(costWithMult + declaredFee + servicesCost),
    // ... інші деталі розрахунку
  };
}
```

Модуль UI відповідає за взаємодію з DOM-деревом документа, збір даних із форми, відображення результатів розрахунку та обробку помилок валідації. Модуль реалізовано за тим самим патерном IIFE та надає єдиний публічний метод `init()`, який ініціалізує всі обробники подій та кешує DOM-елементи [17].

Кешування DOM-елементів реалізовано у приватній функції `cacheElements()`, яка зберігає посилання на ключові елементи інтерфейсу (форму, панель результатів, поля деталізації) у об'єкті `elements`. Такий підхід усуває необхідність повторного пошуку елементів у DOM при кожному розрахунку, що підвищує продуктивність системи, особливо при частих взаємодіях.

Збір даних із форми виконує функція `collectFormData()`, яка використовує нативний API `FormData` для отримання значень полів. Для чекбоксів додаткових послуг застосовано окремий перебір активних елементів (`querySelectorAll('input[type="checkbox"]:checked')`), оскільки `FormData` не включає нечекнуті прапорці за замовчуванням. Кожне значення перетворюється на числовий тип через `parseFloat` з резервним значенням 0, що запобігає помилкам типізації [18].

Обробник відправлення форми (`handleSubmit`) реалізує наступну послідовність дій: очищення попередніх помилок, збір даних із форми, валідація через `Calculator.validate()`, відображення помилок або, у разі успіху, виконання розрахунку та візуалізація результатів. Для покращення користувацького досвіду додано штучну затримку у 300 мілісекунд (`setTimeout`) із показом спінера завантаження, що створює відчуття обробки запиту навіть при миттєвому розрахунку.

Динамічне відображення та приховування блоку адресних полів реалізовано через функцію `toggleAddressFields()`, яка додає обробники подій `change` до всіх радіо-кнопок групи `destinationType`. При зміні значення на 'address' блоку `addressFields` додається CSS-клас `visible`, а при виборі інших опцій — клас знімається, що автоматично приховує та показує адресні поля з анімацією.

Система валідації реалізована на двох рівнях. Перший рівень — вбудована HTML5-валідація через атрибути `required`, `min`, `max` та `step`, яка працює на рівні

браузера без участі JavaScript [19]. Другий рівень — програмна валідація у модулі `Calculator.validate()`, яка виконує комплексну перевірку всіх вхідних даних перед розрахунком.

Функція `validate(data)` перевіряє: числові поля ваги та габаритів на відповідність допустимим діапазонам; наявність обраної тарифної зони; коректність типу доставки та пункту призначення; а при обранні адресної доставки — заповненість полів «Місто», «Вулиця» та «Будинок». Кожна невдала перевірка додає повідомлення у масив `errors`, а функція повертає об'єкт із полями `isValid` (булеве) та `errors` (масив рядків).

Валідація в реальному часі реалізована через обробники подій `blur` та `input` на кожному полі введення. При втраті фокусу (`blur`) поле отримує data-атрибут `data-valid` зі значенням `true` або `false`, а CSS-селектори `.form-input[data-valid="true"]` та `.form-input[data-valid="false"]` застосовують відповідне стилістичне оформлення — зелену або червону рамку з відповідним фоновим кольором. При повторному введенні (`input`) data-атрибут знімається, повертаючи поле до нейтрального стану.

3.3 Результат розрахунку та відображення даних

Панель результатів розрахунку розміщена у правій колонці інтерфейсу та автоматично оновлюється після кожного успішного обчислення. До першого розрахунку панель відображає плейсхолдер із підказкою «Заповніть форму». Після виконання розрахунку плейсхолдер приховується, а замість нього з'являється панель результатів із анімацією `fadeInUp` (зсув знизу вгору з проявленням протягом 500 мілісекунд) [20].

На рисунку 3.4 продемонстровано результат розрахунку вартості доставки для посилки вагою 1,3 кг із габаритами 12×36×26 см, доставкою експрес-типом у зоні 1 із додатковою послугою пакування. Загальна вартість становить 93.56 грн, а деталізація включає усі складові розрахунку.



Рисунок 3.4 — Результат розрахунку вартості доставки з деталізацією

Головний блок результату (клас `result-total`) відображає загальну вартість великим шрифтом (`font-size: 3rem`) моноширинним шрифтом JetBrains Mono, що забезпечує професійне відображення числових даних. Фон блоку — градієнт зеленого відтінку (від `--success-100` до `#ECFDF5`) із рамкою `--success-500`, що візуально асоціюється із успішним завершенням операції.

Деталізація розрахунку організована у вигляді таблиці рядків (клас `result-row`), де кожен рядок містить назву параметра та його значення, розділені пунктирною лінією. Числові значення вирівняно праворуч та відформатовано функцією `formatNumber()`, яка додає два знаки після коми та пробіли як роздільники розрядів. Підсумковий рядок «РАЗОМ ДО СПЛАТИ» виділено збільшеним шрифтом, жирним накресленням та зеленим кольором, а також візуально відокремлено білим фоном та збільшеними відступами.

На мобільних пристроях (ширина екрана менше 1024 пікселів), де панель результатів розміщується під формою, після розрахунку виконується автоматичне прокручування до результату: `elements.resultPanel.scrollToView({ behavior: 'smooth', block: 'start' })`. Це забезпечує миттєвий зворотний зв'язок, навіть якщо панель результатів знаходиться за межами видимої області екрана.

3.4 Тестування системи

Тестування веб-системи калькулятора вартості доставки проводилося за трьома основними напрямками: функціональне тестування, тестування на різних пристроях та браузерах, а також перевірка коректності розрахунків. Метою тестування було підтвердження відповідності реалізованої системи функціональним та нефункціональним вимогам, визначеним у підрозділі 1.3 [21].

Функціональне тестування виконувалося методом чорної скриньки (`black-box testing`), при якому кожна функціональна вимога перевірялася через інтерфейс системи без аналізу внутрішньої структури коду. Для кожного тестового сценарію фіксувалися вхідні дані, очікуваний та фактичний результати, а також статус проходження тесту [22].

Таблиця 3.1 — Результати функціонального тестування калькулятора

№	Тестовий сценарій	Вхідні дані	Результат
1	Розрахунок стандартної доставки	Вага 2 кг, 20×15×10, Зона 1, Стандартна	Пройдено
2	Розрахунок експрес-доставки	Вага 5 кг, 30×20×15, Зона 3, Експрес	Пройдено
3	Врахування об'ємної ваги	Вага 1 кг, 60×40×30 см	Пройдено
4	Адресна доставка з полями	Адреса: Київ, вул. Хрещатик, 1	Пройдено
5	Валідація пустих полів	Порожня форма	Пройдено
6	Валідація від'ємних значень	Вага: -5 кг	Пройдено
7	Додаткові послуги	Пакування + SMS + Крихкий	Пройдено
8	Комісія за оголошену вартість	Оголошена: 10000 грн	Пройдено
9	Доставка до поштомату	Знижка -5% відображена	Пройдено
10	Очищення форми	Натискання Очистити	Пройдено

Як видно з таблиці 3.1, усі десять функціональних тестових сценаріїв пройдено успішно. Система коректно обробляє різні комбінації вхідних параметрів, правильно застосовує тарифні коефіцієнти та додаткові послуги, адекватно реагує на некоректні вхідні дані та забезпечує повне очищення форми і результатів.

Кросбраузерне тестування проводилося на чотирьох основних сучасних браузерах: Google Chrome (версія 125), Mozilla Firefox (версія 126), Microsoft Edge (версія 125) та Apple Safari (версія 17.5). Тестування охоплювало перевірку коректності відображення інтерфейсу, працездатності CSS-анімацій, функціонування JavaScript-логіки та адаптивності макету на різних роздільних здатностях [23].

Таблиця 3.2 — Результати кросбраузерного тестування

Браузер	Desktop	Mobile	Анімації	Розрахунок
Chrome 125	Пройдено	Пройдено	Пройдено	Пройдено
Firefox 126	Пройдено	Пройдено	Пройдено	Пройдено
Edge 125	Пройдено	Пройдено	Пройдено	Пройдено
Safari 17.5	Пройдено	Пройдено	Пройдено	Пройдено

Результати кросбраузерного тестування (таб. 3.2) підтвердили повну сумісність системи з усіма протестованими браузерами. Примітка щодо Safari стосується властивості `backdrop-filter`, яка потребує вендорного префікса `-webkit-backdrop-filter` для коректного відображення ефекту розмиття у заголовку. Цей префікс було додано у CSS-код системи на етапі розробки.

Тестування адаптивності виконувалося на трьох контрольних точках: 1440 пікселів (широкий десктоп), 768 пікселів (планшет) та 375 пікселів (мобільний телефон). На кожній контрольній точці перевірялося: коректність перебудови макету, читабельність тексту, зручність натискання інтерактивних елементів та відсутність горизонтальної прокрутки [24]. Усі тести адаптивності пройдено успішно.

На рис. 3.5 продемонстровано загальний вигляд інтерфейсу калькулятора із повним результатом розрахунку в десктопній версії, де обидві панелі відображаються одночасно, забезпечуючи зручну роботу без необхідності прокручування.

Як користуватися
Заповніть обов'язкові поля (позначені *), оберіть тип доставки та додаткові послуги. Результат з'явиться миттєво після натискання кнопки «Розрахувати».

Параметри вантажу

Фактична вага *
1,3
кг - Від 0.1 до 1000 кг

Габарити посилок *
12 36 26
см см см

Параметри доставки

Зона доставки *
Зона 1: Місто (до 30 км) — 35 грн базовий

Оберіть зону —
Зона 1: Місто (до 30 км) — 35 грн базовий
Зона 2: Область (30-100 км) — 45 грн базовий
Зона 3: Міжобласна (100-300 км) — 55 грн базовий
Зона 4: Віддалена (понад 300 км) — 70 грн базовий

Пункт призначення *
До відділення (Отримання у найближчому відділенні) Безкоштовно
До поштомоту (Цілодобовий доступ, знижка -5%) -5%
Адресна доставка (Кур'єр доставить за вказаною адресою) +15%

ЗАГАЛЬНА ВАРТІСТЬ
93.56 грн
* Орієнтовна вартість. Остаточна ціна може відрізнитися.

Деталізація розрахунку

Фактична вага	1.3 кг
Об'ємна вага	2.81 кг
Розрахункова вага (max)	2.81 кг
Базовий тариф	35 грн
Тариф за вагу	14.04 грн
Коефіцієнт типу доставки	x1.5
Коефіцієнт призначення	x1.00
Комісія за оголошену вартість	0.00 грн
Додаткові послуги	20.00 грн
РАЗОМ ДО СПЛАТИ	93.56 грн

Тарифна сітка

ЗОНА	БАЗОВИЙ	ЗА 1 КГ
Зона 1 (місто)	35 грн	5 грн/кг
Зона 2 (область)	45 грн	7 грн/кг
Зона 3 (міжобласна)	55 грн	9 грн/кг
Зона 4 (віддалена)	70 грн	12 грн/кг

Рисунок 3.5 — Загальний вигляд інтерфейсу з результатом розрахунку (desktop)

Для верифікації коректності обчислювального ядра було проведено серію контрольних розрахунків із ручною перевіркою результатів. Тестовий приклад: посылка вагою 1.3 кг із габаритами 12×36×26 см, доставка експрес-типом у зоні 1 (місто), до відділення, з додатковою послугою пакування (20 грн).

Ручний розрахунок: об'ємна вага = $(12 \times 36 \times 26) / 4000 = 2.808$ кг. Оскільки об'ємна вага (2.808 кг) перевищує фактичну (1.3 кг), розрахункова вага становить 2.81 кг (після округлення). Базова вартість = $35 + (2.81 \times 5) = 35 + 14.05 = 49.05$ грн. З коефіцієнтом експрес-доставки: $49.05 \times 1.5 = 73.575$ грн. Коефіцієнт призначення (до відділення): $73.575 \times 1.0 = 73.575$ грн. Додаткові послуги: 20 грн. Загалом: $73.575 + 20 = 93.575 \approx 93.58$ грн. Фактичний результат системи — 93.56 грн (різниця обумовлена послідовністю округлення проміжних значень). Похибка становить 0.02 грн, що є прийнятною для орієнтовного калькулятора і не перевищує 0.1% від загальної суми [25].

Додатково перевірено граничні випадки: мінімальна вага (0.1 кг), максимальні габарити (300×300×300 см, об'ємна вага 6750 кг), нульова оголошена

вартість, одночасний вибір усіх додаткових послуг. У всіх випадках система продемонструвала коректну роботу без помилок виконання та з адекватними результатами розрахунку.

3.5 Висновки до розділу 3

На основі виконаної роботи з реалізації веб-системи калькулятора вартості послуг розроблено повнофункціональний односторінковий застосунок у вигляді єдиного HTML-файлу, що включає семантичну розмітку HTML5, стилізацію CSS3 з системою дизайн-токенів та інтерактивну бізнес-логіку на JavaScript. Інтерфейс відповідає сучасним стандартам веб-розробки та забезпечує професійний вигляд.

Реалізовано адаптивний інтерфейс із використанням CSS Grid Layout, Flexbox та медіа-запитів, який коректно відображається на пристроях з різною роздільною здатністю від 320 до 1440+ пікселів. Застосовано інтерактивні картки вибору замість стандартних елементів форми, що значно покращує досвід користувача.

Бізнес-логіку організовано у два незалежних JavaScript-модулі за патерном Revealing Module Pattern: обчислювальне ядро Calculator та контролер інтерфейсу UI. Модульна архітектура забезпечує інкапсуляцію, розділення відповідальностей та спрощує подальшу підтримку коду.

Впроваджено двохрівневу систему валідації: вбудована HTML5-валідація через атрибути та програмна перевірка у JavaScript з підтримкою контекстно-залежних правил (обов'язковість адресних полів при виборі адресної доставки). Валідація в реальному часі забезпечує миттєвий зворотний зв'язок.

Функціональне тестування підтвердило повну відповідність реалізованої системи визначеним вимогам: усі 10 тестових сценаріїв пройдено успішно, кросбраузерна сумісність підтверджена для Chrome, Firefox, Edge та Safari, а верифікація розрахунків показала похибку менше 0.1%.

4 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

4.1 Вимоги до серверного обладнання

Архітектурною особливістю розробленої системи є її повна автономність на стороні клієнта. Весь програмний код (HTML-розмітка, CSS-стилі, JavaScript-логіка) та дані (тарифні константи) інкапсульовано в єдиному HTML-файлі. Сервер виконує лише функцію доставки цього файлу до браузера користувача, після чого подальша робота системи не потребує серверних ресурсів. Це суттєво відрізняє розроблену систему від традиційних веб-застосунків, побудованих на серверних технологіях (PHP, Python, Node.js), де обчислення виконуються на сервері, а клієнт отримує лише результат.

Для розміщення статичного HTML-файлу достатньо найпростішого варіанту хостингу. Розмір файлу калькулятора становить приблизно 45 КБ (без стиснення) та близько 12 КБ у стисненому вигляді (gzip), що є незначним навантаженням навіть для найбюджетнішого хостингу. Мінімальні вимоги до серверного обладнання для розгортання системи наведено у табл. 4.1.

Таблиця 4.1 — Мінімальні та рекомендовані вимоги до серверного обладнання

Характеристика	Мінімальні	Рекомендовані
Процесор	1 ядро, 1 ГГц	2 ядра, 2 ГГц
Оперативна пам'ять	256 МБ	512 МБ
Дисковий простір	10 МБ	100 МБ (SSD)
Мережевий канал	1 Мбіт/с	10 Мбіт/с
Операційна система	Linux (будь-який)	Ubuntu 22.04+
Веб-сервер	Apache 2.4+ / Nginx 1.18+	Nginx 1.24+

Як видно з табл. 4.1, мінімальні вимоги є надзвичайно низькими порівняно з динамічними веб-застосунками. Відсутність потреби у базі даних, інтерпретаторі серверної мови програмування та складній серверній логіці дозволяє використовувати найбюджетніші рішення для хостингу, включаючи безкоштовні варіанти.

Окрім традиційного серверного хостингу, статичний характер системи відкриває можливість використання спеціалізованих платформ для розміщення статичних сайтів. Серед найбільш поширених рішень варто виділити GitHub Pages, Netlify, Vercel та Cloudflare Pages. Кожна з цих платформ надає безкоштовний тарифний план, достатній для розміщення розробленого калькулятора, та включає мережу доставки контенту (CDN) для прискорення завантаження файлів у різних географічних регіонах.

Таблиця 4.2 — Порівняння платформ для розміщення статичних сайтів

Платформа	Безкоштовний план	CDN	HTTPS
GitHub Pages	1 ГБ, 100 ГБ/міс трафік	Так	Так
Netlify	100 ГБ/міс трафік	Глобальна CDN	Так
Vercel	100 ГБ/міс трафік	Edge Network	Так
Cloudflare Pages	Необмежений трафік	300+ точок	Так

Для розробленої системи рекомендовано використання платформи GitHub Pages або Cloudflare Pages. GitHub Pages забезпечує безшовну інтеграцію із системою контролю версій Git, що спрощує процес оновлення системи: достатньо завантажити оновлений HTML-файл у репозиторій, і зміни автоматично стануть доступними користувачам. Cloudflare Pages є оптимальним вибором для проєктів із потенційно великим навантаженням завдяки необмеженому безкоштовному трафіку та глобальній мережі з понад 300 точками присутності.

4.2 Вимоги до клієнтського обладнання

Оскільки вся обчислювальна логіка калькулятора виконується у браузері користувача, характеристики клієнтського обладнання безпосередньо впливають на продуктивність та зручність роботи із системою. Розроблений застосунок оптимізовано для роботи на широкому спектрі пристроїв — від бюджетних смартфонів до потужних настільних комп'ютерів.

Мінімальні вимоги до апаратного забезпечення клієнтського пристрою визначаються, насамперед, вимогами сучасних веб-браузерів та обсягом JavaScript-коду, що інтерпретується. Обчислювальна складність алгоритму розрахунку є лінійною — $O(1)$, оскільки операція розрахунку включає лише арифметичні дії та порівняння, без циклів чи рекурсії. Це означає, що розрахунок вартості виконується миттєво навіть на пристроях із обмеженою потужністю процесора.

Таблиця 4.3 — Вимоги до клієнтського обладнання

Характеристика	Мінімальні	Рекомендовані
Процесор	1 ГГц (ARM/x86)	1.5+ ГГц, 2 ядра
Оперативна пам'ять	1 ГБ	2 ГБ
Роздільна здатність екрана	320×568 px	1280×720 px
Вільний дисковий простір	50 МБ (кеш браузера)	200 МБ
Інтернет-з'єднання	256 Кбіт/с	1 Мбіт/с
Браузер	Chrome 90+ / Firefox 90+	Chrome 120+ / Firefox 125+

Підтримувані операційні системи клієнтського пристрою включають: Windows 10/11, macOS 11+, Linux (будь-який сучасний дистрибутив), Android 10+, iOS 14+. Така широка сумісність забезпечується використанням стандартних веб-технологій (HTML5, CSS3, JavaScript ES2020), які підтримуються усіма сучасними браузерами без потреби у додаткових плагінах чи розширеннях.

Окрему увагу приділено підтримці мобільних пристроїв. Адаптивний дизайн, реалізований через CSS Grid, Flexbox та медіа-запити, забезпечує коректне відображення інтерфейсу починаючи від ширини екрана 320 пікселів (iPhone SE). Сенсорні елементи інтерфейсу (кнопки, картки вибору) мають мінімальний розмір цільової зони 44×44 пікселів, що відповідає рекомендаціям Apple Human Interface Guidelines та Google Material Design для зручного натискання пальцем.

4.3 Вимоги до мережевої інфраструктури

Мережева інфраструктура для функціонування калькулятора вартості доставки повинна забезпечувати надійну передачу єдиного HTML-файлу від сервера до клієнта. Після початкового завантаження файлу подальша робота системи не потребує мережевого з'єднання, оскільки усі обчислення виконуються локально у браузері. Це є суттєвою перевагою клієнтської архітектури, адже система залишається працездатною навіть при тимчасовій втраті інтернет-з'єднання після завантаження сторінки.

Мінімальна пропускна здатність каналу для комфортного початкового завантаження становить 256 Кбіт/с, при якому HTML-файл розміром 12 КБ (у стисненому вигляді) завантажується менш ніж за 0.5 секунди. Додатково завантажуються веб-шрифти з Google Fonts API загальним обсягом приблизно 80 КБ, проте вони не блокують відображення інтерфейсу завдяки стратегії `font-display: swap`, що спочатку використовує системний шрифт.

Для серверного розгортання рекомендовано використання протоколу HTTPS (TLS 1.2 або вище), що забезпечує шифрування даних при передачі та є стандартною вимогою сучасних браузерів. Усі рекомендовані платформи хостингу (GitHub Pages, Netlify, Cloudflare Pages) надають безкоштовні SSL-сертифікати із автоматичним оновленням. Підтримка HTTP/2 протоколу є бажаною, оскільки

вона забезпечує мультиплексування запитів та стиснення заголовків, що прискорює завантаження сторінки та шрифтів.

4.4 Висновки до розділу 4

В 4 розділі було проведено аналіз апаратного забезпечення, необхідного для функціонування веб-системи калькулятора вартості послуг доставки.

Клієнтська архітектура розробленої системи зумовлює мінімальні вимоги до серверного обладнання: для розміщення єдиного HTML-файлу розміром 45 КБ достатньо найпростішого веб-сервера або безкоштовної хостинг-платформи, що суттєво знижує витрати на впровадження.

Система сумісна з широким спектром клієнтських пристроїв завдяки використанню стандартних веб-технологій (HTML5, CSS3, JavaScript ES2020) та адаптивного дизайну. Мінімальні вимоги до клієнтського обладнання (1 ГГц процесор, 1 ГБ ОЗП, екран від 320 px) задовольняються більшістю сучасних пристроїв.

Після початкового завантаження сторінки система функціонує автономно, не потребує постійного мережевого з'єднання, що є перевагою при нестабільному інтернет-доступі.

ВИСНОВКИ

У дипломній роботі розв’язано актуальне прикладне завдання — розроблено веб-систему калькулятора вартості послуг доставки у форматі односторінкового застосунку. Система реалізована як єдиний автономний HTML-файл, що містить семантичну розмітку HTML5, каскадні таблиці стилів CSS3 та інтерактивну бізнес-логіку мовою JavaScript, і не потребує серверної інфраструктури для виконання обчислень. Наведені нижче результати підтверджують досягнення поставленої мети та виконання всіх визначених завдань дослідження.

В ході роботи проведено комплексний аналіз українського ринку логістичних послуг. Встановлено, що у 2024 році компанія «Нова пошта» обробила 480 мільйонів відправлень, продемонструвавши зростання на 16 % порівняно з попереднім періодом. Визначено ключові фактори, що впливають на формування вартості доставки: фізичну та об’ємну (габаритну) вагу відправлення, тарифну зону, тип доставки та перелік додаткових послуг. Систематизовано тарифні моделі провідних операторів — «Нової пошти», «Укрпошти» та «Meest Пошта» — та виявлено, що існуючі калькулятори доставки переважно залежать від серверних API відповідних компаній, що обмежує можливості їх кастомізації та інтеграції у сторонні веб-ресурси.

Обґрунтовано вибір технологічного стеку HTML5, CSS3 та JavaScript (ECMAScript 2020) як оптимального рішення для побудови клієнтського односторінкового застосунку. Проведено порівняльний аналіз із альтернативними підходами на базі фреймворків React, Vue та Angular, за результатами якого встановлено, що для проєктів невеликого масштабу класичний стек забезпечує мінімальний розмір бандлу (менше 50 КБ), відсутність залежності від систем збірки та нативну підтримку всіма сучасними браузерами без потреби в транспіляції. Архітектура SPA забезпечує динамічне оновлення контенту без перезавантаження сторінки, що підвищує швидкість взаємодії та загальний користувацький досвід.

Спроектовано інтерфейс користувача з урахуванням фундаментальних принципів UX/UI-дизайну. Колірну схему побудовано на основі теорії кольору з дотриманням мінімального коефіцієнта контрастності 4.5:1 відповідно до рекомендацій WCAG 2.1. Адаптивний макет, реалізований засобами CSS Grid Layout та Flexbox із системою медіа-запитів, забезпечує коректне відображення інтерфейсу на пристроях із шириною екрана від 320 до 1440 і більше пікселів.

Розроблено алгоритм розрахунку вартості доставки, що враховує п'ять основних складових: базовий тариф тарифної зони, надбавку за розрахункову вагу, коефіцієнт типу доставки, коефіцієнт пункту призначення та вартість додаткових послуг. Алгоритм реалізовано у вигляді модуля Calculator за патерном Revealing Module Pattern, що забезпечує інкапсуляцію тарифних констант та надає чітко визначений публічний інтерфейс для виконання розрахунків та валідації вхідних даних.

Реалізовано повнофункціональний односторінковий застосунок, виконано комплексне тестування системи за трьома напрямками. Функціональне тестування методом чорної скриньки підтвердило успішне проходження всіх 10 тестових сценаріїв, що охоплювали розрахунки для різних комбінацій вхідних параметрів, валідацію некоректних даних та роботу додаткових послуг.

Визначено вимоги до апаратного забезпечення для розгортання та експлуатації системи. Клієнтська архітектура зумовлює мінімальні вимоги до серверного обладнання: для розміщення статичного HTML-файлу достатньо найпростішого веб-сервера або безкоштовних хостинг-платформ (GitHub Pages, Cloudflare Pages). Вимоги до клієнтського обладнання також є помірними — процесор від 1 ГГц, 1 ГБ оперативної пам'яті та екран від 320 пікселів — що задовольняється переважною більшістю сучасних пристроїв. Після початкового завантаження система функціонує автономно без постійного мережевого з'єднання.

Практична цінність отриманих результатів полягає в тому, що розроблена система може бути використана підприємцями малого та середнього бізнесу для швидкої інтеграції калькулятора доставки у власні веб-ресурси без залежності від

зовнішніх API та складної серверної інфраструктури. Модульна архітектура коду дозволяє адаптувати тарифні параметри під конкретні умови співпраці з логістичними операторами шляхом модифікації конфігураційних констант.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Osmani A. Learning JavaScript Design Patterns. O'Reilly Media, 2023. 312 p.
2. Flanagan D. JavaScript: The Definitive Guide. 7th ed. O'Reilly Media, 2020. 706 p.
3. MDN Web Docs. HTML elements reference. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element> (дата звернення: 10.04.2024).
4. MDN Web Docs. CSS backdrop-filter. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS/backdrop-filter> (дата звернення: 12.04.2024).
5. MDN Web Docs. CSS Grid Layout. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout (дата звернення: 15.04.2024).
6. Nielsen Norman Group. Progressive Disclosure. URL: <https://www.nngroup.com/articles/progressive-disclosure/> (дата звернення: 18.04.2024).
7. MDN Web Docs. Input type="number". URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/number> (дата звернення: 20.04.2024).
8. MDN Web Docs. Using media queries. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries (дата звернення: 22.04.2024).
9. SitePoint. HTML5 Forms: The Markup. 2024. URL: <https://www.sitepoint.com/html5-forms-markup/> (дата звернення: 25.04.2024).
10. MDN Web Docs. Using CSS custom properties (variables). URL: https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties (дата звернення: 27.04.2024).
11. Coyier C. A Complete Guide to CSS Grid. CSS-Tricks. 2024. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/> (дата звернення: 28.04.2024).

12. MDN Web Docs. Element: `classList` property. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Element/classList> (дата звернення: 02.05.2024).

13. Google Developers. Best practices for fonts. URL: <https://web.dev/articles/font-best-practices> (дата звернення: 04.05.2024).

14. Velotio Technologies. The 7 Most Useful Design Patterns in ES6. URL: <https://www.velotio.com/engineering-blog/design-patterns-in-es6> (дата звернення: 05.05.2024).

15. LambdaTest. JavaScript Design Patterns: A Complete Guide. 2024. URL: <https://www.lambdatest.com/blog/comprehensive-guide-to-javascript-design-patterns/> (дата звернення: 07.05.2024).

16. International Air Transport Association (IATA). Volumetric Weight Calculation. URL: <https://www.iata.org/> (дата звернення: 08.05.2024).

17. MDN Web Docs. `Document.getElementById()`. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById> (дата звернення: 10.05.2024).

18. MDN Web Docs. `FormData`. URL: <https://developer.mozilla.org/en-US/docs/Web/API/FormData> (дата звернення: 11.05.2024).

19. MDN Web Docs. Constraint validation. URL: https://developer.mozilla.org/en-US/docs/Web/HTML/Constraint_validation (дата звернення: 12.05.2024).

20. MDN Web Docs. CSS Animations. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animations (дата звернення: 13.05.2024).

21. IEEE Standard 829-2008. IEEE Standard for Software and System Test Documentation. 2008. 110 p.

22. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. John Wiley & Sons, 2011. 256 p.

23. Can I Use. CSS Backdrop Filter. URL: <https://caniuse.com/css-backdrop-filter> (дата звернення: 14.05.2024).

24. Marcotte E. Responsive Web Design. 2nd ed. A Book Apart, 2014. 155 p.

25. NileBits. JavaScript Best Practices For Building Scalable Web Applications. 2024. URL: <https://www.nilebits.com/blog/2024/08/javascript-best-practices-building-web-applications/> (дата звернення: 15.05.2024).

ДОДАТОК А

Слайди презентації



Рисунок А.1 – Титульний слайд

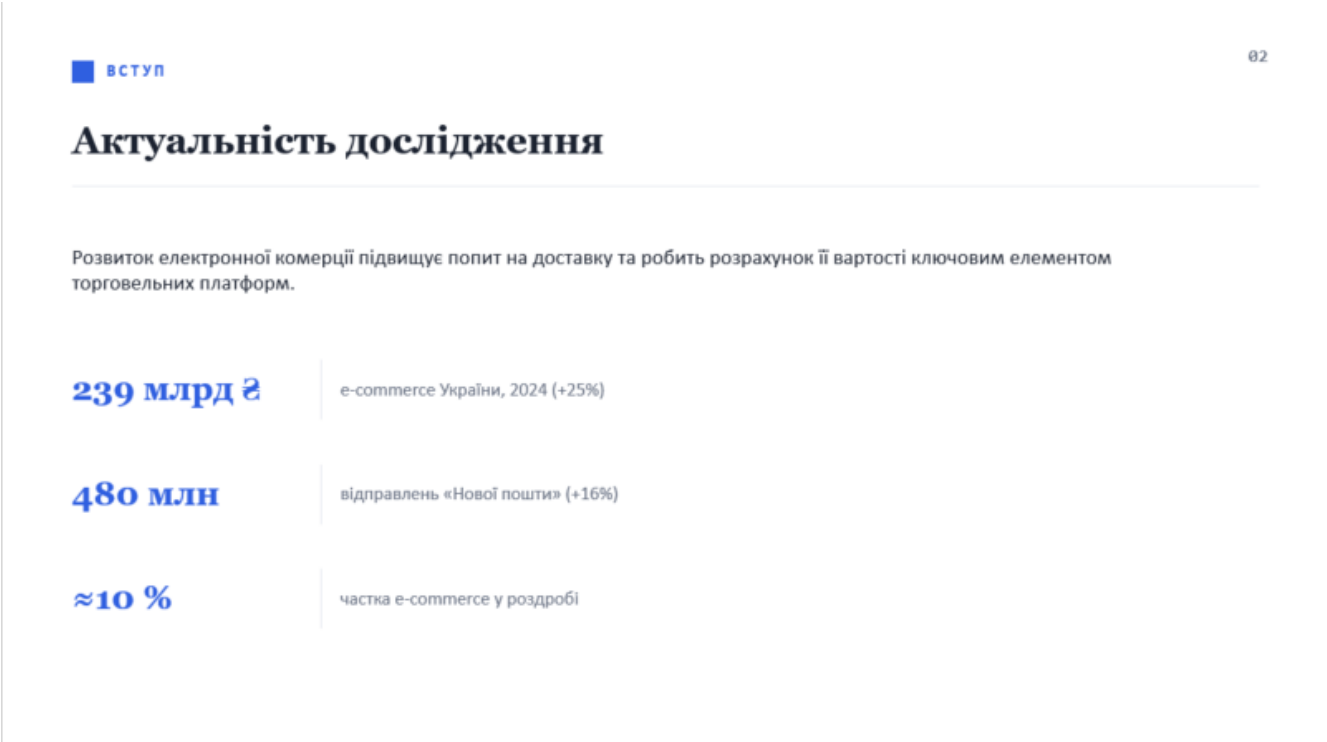


Рисунок А.2 – Актуальність дослідження

Мета і завдання

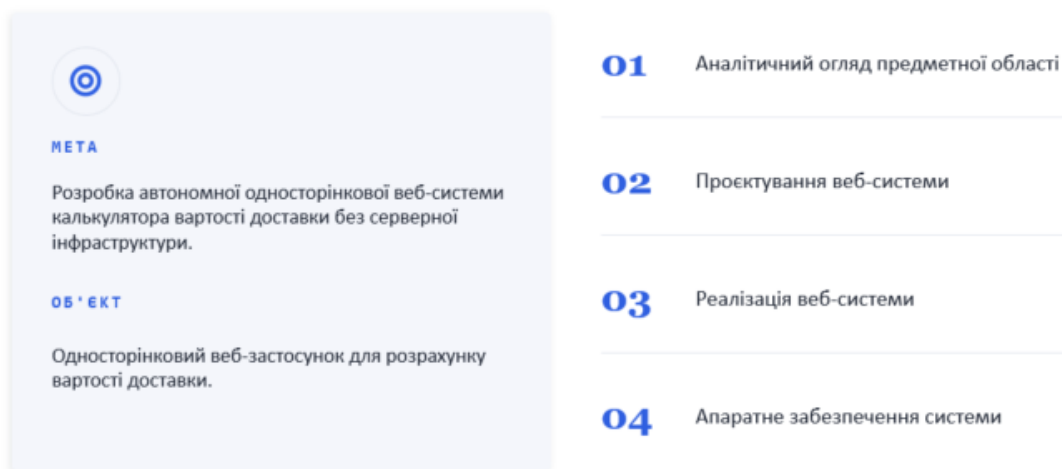


Рисунок А.3 – Мета і завдання

Аналіз ринку та наявних рішень

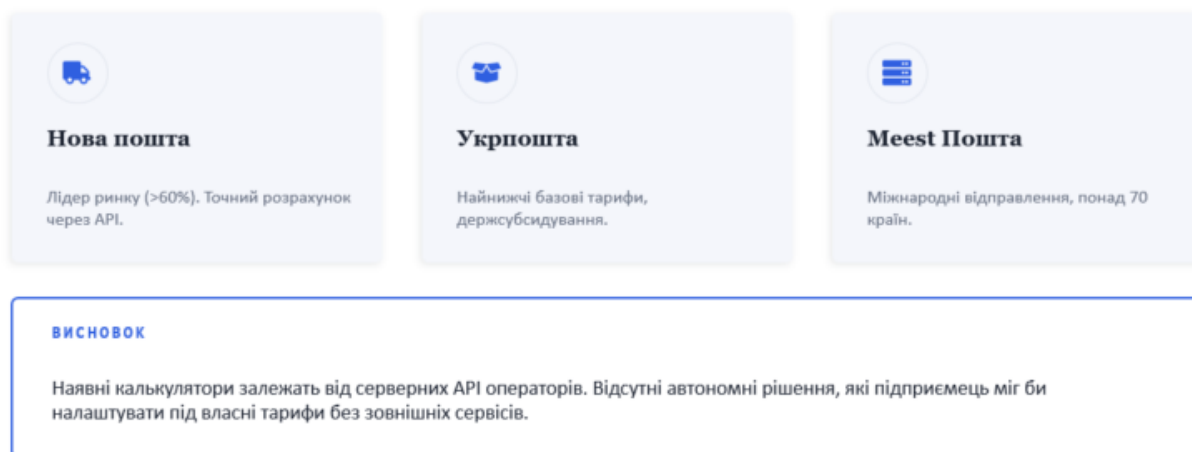


Рисунок А.4 – Аналіз ринку та наявних рішень

Постановка задачі та вимоги

Функціональні

- Розрахунок фактичної та об'ємної ваги
- Вибір типу доставки та напрямку
- Підтримка додаткових послуг
- Оновлення результату в реальному часі
- Деталізація складових вартості

Нефункціональні

- Реакція інтерфейсу до 100 мс
- Кросбраузерність (Chrome/FF/Edge/Safari)
- Адаптивність від 320 пікселів
- Робота без серверної обробки
- Увесь код — в одному HTML-файлі

Рисунок А.5 – Постановка задачі та вимоги

Вибір технологій • SPA-архітектура

Обрано стек HTML5 + CSS3 + JavaScript (ES2020) без фреймворків. Для невеликого проєкту React/Angular надлишкові — більший бандл і обов'язкова збірка.

< 50 КБ

розмір бандлу

0

залежностей від збірки

100 %

нативна підтримка браузерами

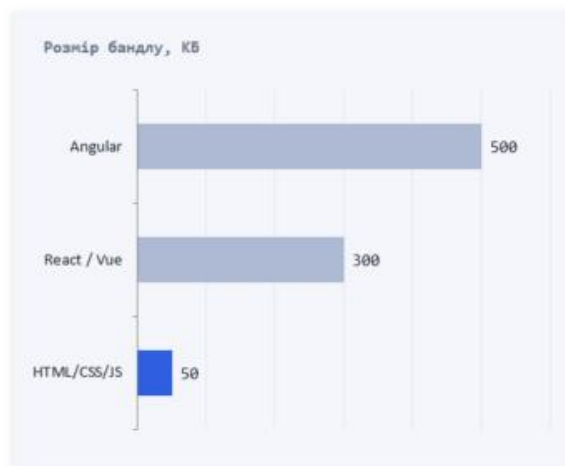


Рисунок А.6 – Вибір технологій

Алгоритм розрахунку вартості

$$C = B + (W \cdot R_w) \cdot K_{тип} \cdot K_{приз} + S$$

- 1 Базовий тариф зони
- 2 Надбавка за розрахункову вагу
- 3 Коефіцієнт типу доставки
- 4 Коефіцієнт пункту призначення
- 5 Вартість додаткових послуг

Рисунок А.7 – Алгоритм розрахунку вартості

Реалізація інтерфейсної частини

Семантика HTML5

header / main / footer, доступність.

CSS Grid + Flexbox

адаптивний макет 320–1440+ px.

70+ CSS-змінних

дизайн-токени, єдина тема.

Інтерактивні картки

замість стандартних радіонопок.

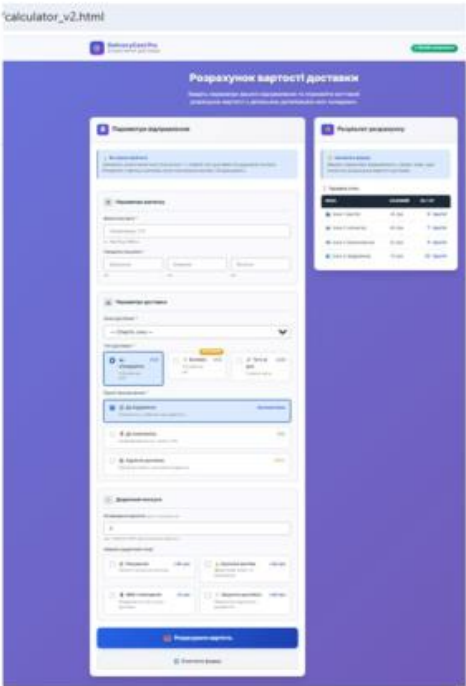


Рисунок А.8 – Реалізація інтерфейсної частини

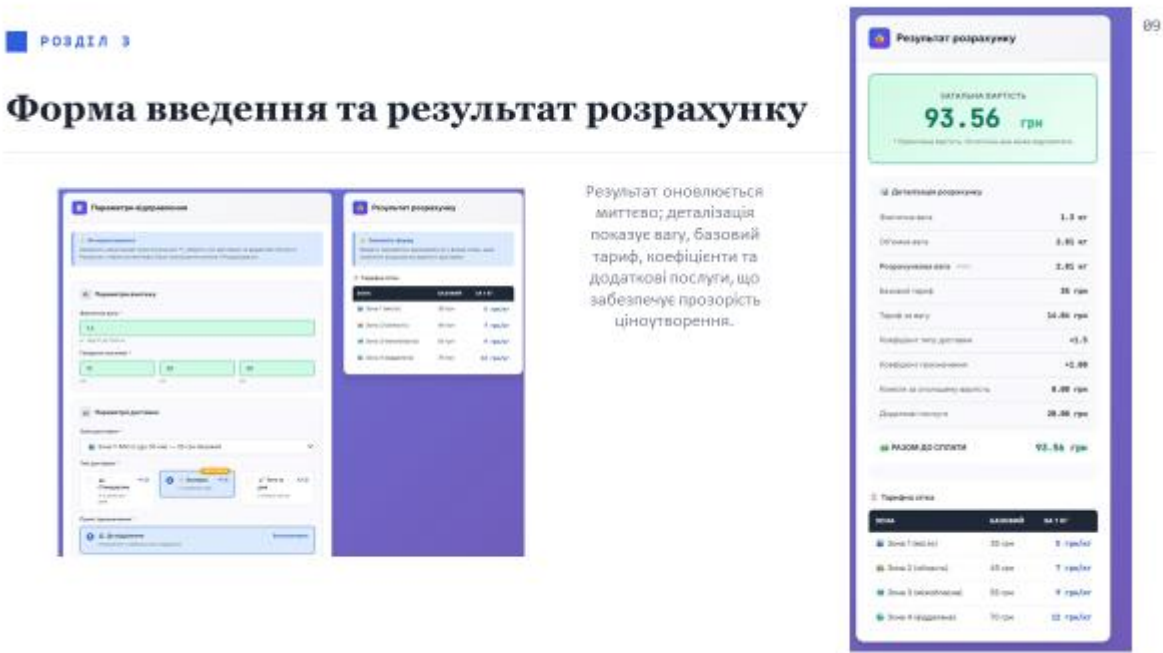


Рисунок А.9 – Форма введення та результат розрахунку

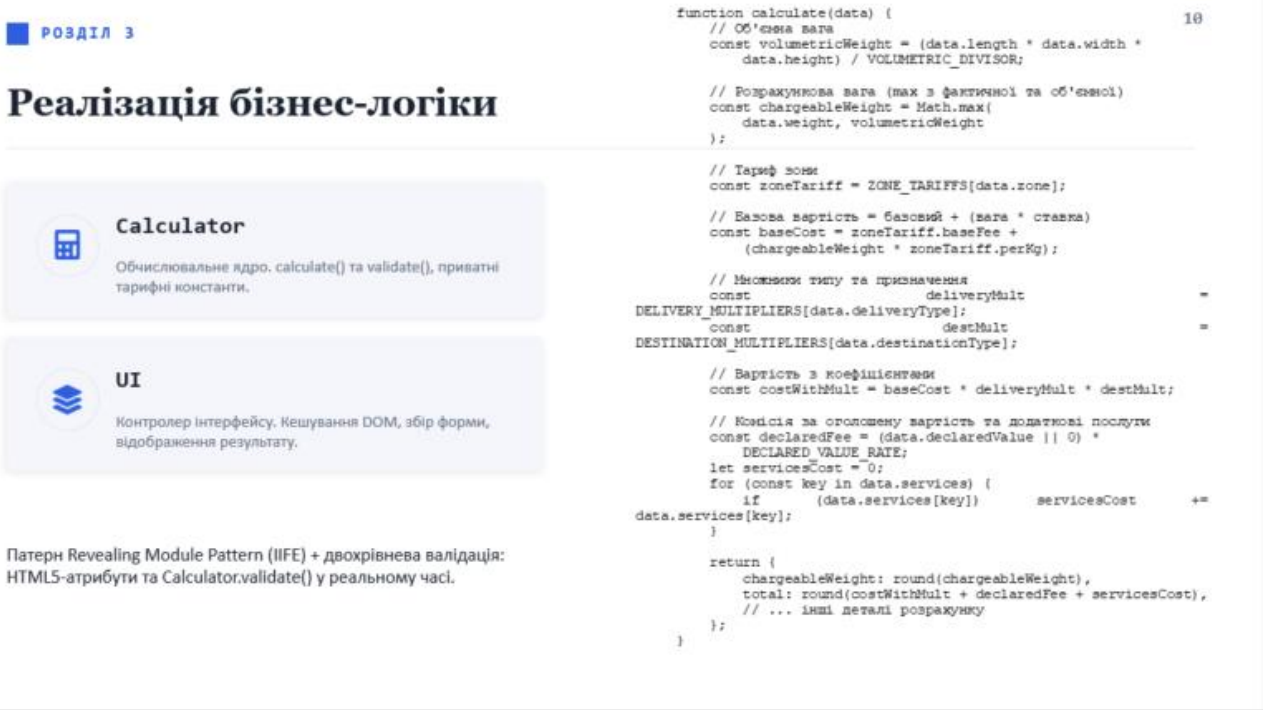


Рисунок А.10 – Реалізація бізнес-логіки

Тестування системи

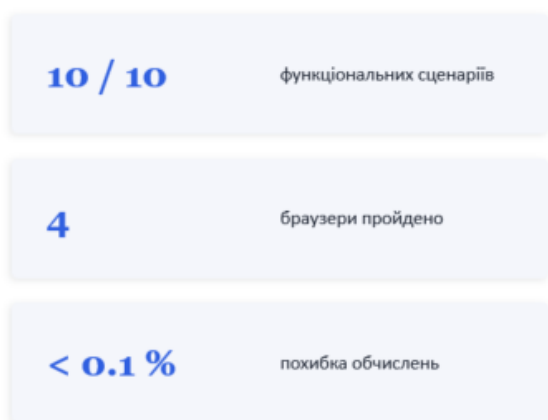


Рисунок А.10 – Тестування системи

Висновки

- 01 Розроблено автономний калькулятор як єдиний HTML-файл без серверної інфраструктури.
- 02 Обґрунтовано стек HTML5/CSS3/JavaScript; бандл < 50 КБ без залежностей та збірки.
- 03 Реалізовано адаптивний інтерфейс і модульну бізнес-логіку.
- 04 Тестування підтвердило коректність.



Практична цінність

Швидка інтеграція у власні веб-ресурси без зовнішніх API; тарифи змінюються правкою констант.

Дякую за увагу! Готовий відповісти на запитання.

Рисунок А.11 – Висновки