

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка сайту з розрахунку вартості металопластикових конструкцій
на основі мови JS

Виконав(ла): студент(ка) 4 курсу, групи СП-41
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис)

Кіндзерський Н. І.

(прізвище та ініціали)

Керівник

(підпис)

Стефанишин В. М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Стоянов Ю. М.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М. Р.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет _____
(повна назва факультету)

Кафедра _____
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) _____
(прізвище та ініціали)
« » 20__р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня _____
(назва освітнього ступеня)

за спеціальністю _____
(шифр і назва спеціальності)

студенту _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____

Керівник роботи _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20__року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

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

6. Консультанти розділів роботи

[illegible]

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент _____

(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) _____ (прізвище та ініціали)

РЕФЕРАТ

Дана кваліфікаційна робота призначена для здобуття ступеня бакалавра студента кафедри програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.

Тема: Розробка системи розрахунку вартості металопластикових конструкцій на основі мови JS.

Робота містить: 72 сторінки, 29 рисунків та 7 таблиць.

Головною метою проєкту є створення інструменту, який дозволяє споживачам швидко та зручно отримувати інформацію про доступні металопластикові вироби, а також обчислювати їхню вартість залежно від обраних параметрів.

Проєкт спрямований на підвищення прозорості цінової політики, полегшення процесу вибору та замовлення продукції для клієнтів, а також на оптимізацію роботи продавців і консультантів. Завдяки інтерактивним функціям користувачі можуть налаштовувати конфігурацію виробів відповідно до своїх потреб і бюджетних обмежень, отримуючи миттєві розрахунки вартості та специфікації обраних товарів.

Розроблено каталог, головну сторінку, та калькулятор вартості металопластикових конструкцій використовуючи технології JS.

Перелік ключових слів: металопластикові конструкції, сервер, клієнт, калькулятор, база даних, діаграма прецедентів, веб-сайт, токени.

ANNOTATION

This qualifying work is intended for obtaining a bachelor's degree for a student of the department of software engineering of Ternopil National Technical University named after Ivan Pulyu.

Topic: Development of a system for calculating the cost of metal-plastic structures based on the JS language.

The work contains: 72 pages, 29 figures and 7 tables.

The main goal of the project is to create a tool that allows consumers to quickly and conveniently receive information about available metal-plastic products, as well as calculate their cost depending on the selected parameters.

The project is aimed at increasing the transparency of the price policy, facilitating the process of choosing and ordering products for customers, as well as optimizing the work of salespeople and consultants. Thanks to interactive features, users can customize the configuration of products according to their needs and budget constraints, receiving instant cost calculations and specifications of selected products.

The catalog, the main page, and the calculator of the cost of metal-plastic structures were developed using JS technologies.

Keywords list: metal-plastic structures, server, client, calculator, database, precedent chart, website, tokens.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Актуальність проблеми.....	9
1.2 Аналіз вимог до системи.....	10
1.3 Аналіз конкурентів	11
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	15
2.1 Проєктування варіантів використання системи.....	15
2.2 Проєктування інформаційної системи.....	18
2.3 Проєктування бази даних	21
3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
3.1 Вибір основних технологій веб-розробки	27
3.2 Вибір середовища розробки.....	31
3.3 Реалізація веб-застосунку.....	32
3.4 Тестування програмного забезпечення.....	45
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	51
4.1 Надзвичайні ситуації екологічного характеру.	51
4.2 Суть та зміст управління охороною праці.....	54
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ.....	61
ДОДАТОК А.....	62
ДОДАТОК Б	72

ВСТУП

У сучасному світі металопластикові конструкції, користуються великою популярністю завдяки своїм відмінним експлуатаційним характеристикам, довговічності та енергоефективності. З розвитком будівельних технологій та зростаючими вимогами до енергозбереження попит на такі вироби стабільно зростає. Водночас споживачі стикаються з необхідністю вибору оптимальних конструкцій, що задовольняють їхні потреби як за якістю, так і за ціною.

Ця робота присвячена розробці каталогу та калькулятора вартості металопластикових конструкцій, що має на меті створення зручного і ефективного інструменту для кінцевих споживачів та представників бізнесу.

Головною метою проєкту є спрощення процесу вибору та розрахунку вартості металопластикових конструкцій, що дозволить користувачам швидко знайти необхідні вироби, ознайомитися з їх характеристиками та здійснити точний розрахунок вартості залежно від індивідуальних параметрів і вимог.

Реалізація цих завдань є необхідною умовою для успішного досягнення поставленої мети:

1. Провести аналіз ринку металопластикових конструкцій для визначення основних вимог та побажань користувачів.
2. Розробити архітектуру системи, яка включатиме базу даних для зберігання інформації та інтерфейс користувача для взаємодії з системою.
3. Використати сучасні веб-технології, такі як HTML, CSS, JavaScript, React, BootStrap та Node.js і MongoDB для бекенду, щоб забезпечити високу продуктивність та зручність користування.
4. Реалізувати функціонал калькулятора, який дозволить користувачам обирати необхідні параметри та отримувати миттєвий розрахунок вартості.
5. Провести тестування системи для забезпечення її надійності та точності розрахунків.

Розроблений інструмент дозволяє користувачам налаштовувати конфігурацію виробів відповідно до їхніх потреб і бюджетних обмежень, отримуючи миттєві розрахунки вартості та специфікації обраних товарів. Це підвищує прозорість цінової політики та полегшує процес прийняття рішень, що сприяє покращенню клієнтського досвіду.

Результати цього дослідження мають велике практичне значення. Вони можуть бути використані компаніями, що займаються виробництвом та продажем металопластикових конструкцій, для підвищення якості обслуговування клієнтів та оптимізації процесів продажу. Крім того, кінцеві споживачі отримають можливість здійснювати більш усвідомлений вибір, спираючись на точні та прозорі дані про продукцію.

Таким чином, представлений проект не лише сприяє покращенню взаємодії між компаніями та їх клієнтами, але й підвищує конкурентоспроможність компаній на ринку металопластикових конструкцій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

В сучасний період металопластикові конструкції набувають великої популярності в будівельній і ремонтній сферах завдяки своїм значним перевагам, таким як енергоефективність, довговічність та естетичний вигляд. Однак процес вибору та розрахунку вартості таких конструкцій залишається складним через велику кількість виробників, різноманіття моделей і конфігурацій, а також індивідуальні потреби клієнтів. Застосування сучасних веб-технологій для створення інтерактивних інструментів, таких як каталоги і калькулятори вартості, стає необхідним для полегшення процесу вибору, забезпечення точних розрахунків вартості та зниження часу, необхідного для прийняття рішення.

Металопластикові конструкції актуальні сьогодні з кількох причин.

Застосування каталогу дозволить користувачам швидко знаходити необхідні металопластикові конструкції серед великого асортименту, порівнювати їх характеристики та вибирати оптимальний варіант для свого проекту.

Калькулятор вартості надає можливість отримувати точні розрахунки вартості конструкцій з урахуванням різних параметрів (розмір, тип скла, додаткові опції) і конфігурацій, що забезпечує коректне планування бюджету будівельного проекту.

Це значно знижує час, необхідний для вибору та придбання продукції, забезпечуючи швидкий доступ до необхідної інформації.

Крім того, ці інструменти покращують обслуговування клієнтів, забезпечуючи високий рівень обслуговування і задоволення від покупки завдяки зручному інтерфейсу та доступності інформації.

Отже, розробка інтерактивних каталогів та калькуляторів вартості металопластикових конструкцій є актуальною та важливою задачею, яка відповідає сучасним вимогам ефективності та зручності в обслуговуванні клієнтів.

1.2 Аналіз вимог до системи

Аналіз вимог до програмного забезпечення є критичним етапом розробки проекту, який визначає та документує всі необхідні функціональні та нефункціональні характеристики системи [1]. Цей етап дозволяє чітко зрозуміти очікування користувачів та замовника, а також забезпечити, щоб кінцевий продукт відповідав всім їхнім потребам та вимогам.

Функціональні вимоги:

- Реєстрація та авторизація користувача в системі
- Реалізація модальних вікон
- Відправка введених користувачем даних на сервер
- Фільтрація товарів за категоріями
- Корзина для товарів
- Додавання користувачів в базу даних
- Виведення списку всіх користувачів
- Валідація полів для введення даних
- Реалізація вибору типу вікна, та вибору комплектуючих
- Реалізація калькулятора для обрахування вартості конструкцій

Нефункціональні вимоги:

- Сайт повинен завантажуватися швидко, в межах 3 секунд або менше на основній сторінці.
- Калькулятор повинен повертати результат обчислення миттєво після введення даних користувачем.
- Всі дані користувачів, такі як паролі, мають зберігатися у безпечному форматі.
- Сайт повинен працювати на основних браузерях (Chrome, Firefox, Safari, Edge) та на основних платформах (Windows, macOS, iOS, Android).
- Інтерфейс повинен бути інтуїтивно зрозумілим і легким у використанні для різних категорій користувачів.

1.3 Аналіз конкурентів

На даному ринку присутня велика кількість конкурентів. Проте буде розглянуто лише 2 основних.

Першим з них є VEKA- один з провідних виробників ПВХ-вікон в Україні та світі, з більш ніж 50-річним досвідом у галузі [2]. Компанія спеціалізується на виробництві високоякісних віконних систем, які відповідають всім міжнародним стандартам якості та безпеки. VEKA пропонує детальну інформацію про свою продукцію, включаючи різні типи віконних систем і їх характеристики.

На сайті доступний каталог продукції з можливістю швидкої навігації і вибору необхідного типу вікон. Також є розділ з контактною інформацією, що дозволяє клієнтам швидко зв'язатися з представниками компанії. Сайт пропонує можливість віртуального огляду продукції, що дозволяє користувачам більш детально ознайомитися з виглядом і характеристиками вікон перед покупкою. Це зображено на рис. 1.3.1

Калькулятор вікон ПВХ, м. Чугуїв

Даний інтерфейс передбачає формування заявки, на підставі якої розрахунок буде виконувати менеджер.

Параметри вікна

Ширина, мм: 1400

Висота, мм: 1460

Склопакет: не обраний

Колір: Білий

Обрана конструкція: Одношлюкове вікно

Рисунок 1.3.1 – Система обрахунку вартості

Даний виробник має такі переваги:

- Зрозумілий і легкий у використанні інтерфейс калькулятора, що дозволяє швидко і просто розрахувати вартість металопластикових конструкцій.
- Користувачі можуть надіслати заявку на отримання детальної консультації від кваліфікованих фахівців компанії.
- Завдяки вибору різних опцій, користувачі можуть адаптувати вікна під свої індивідуальні потреби та бюджет.

Проте, цей сервіс має і свої недоліки, такі як:

- Калькулятор має достатньо обмежений вибір опцій порівняно з деякими конкурентами.
- Можливість довгих строків виготовлення та доставки. Через високий попит може бути необхідний більш довгий термін виготовлення і доставки виробу.
- Калькулятор VEKA дозволяє користувачам вибирати різні параметри вікон, але не обраховує підсумкову вартість замовлення, що ускладнює оцінку витрат.

Також одним з найбільш популярних виробників металопластикових конструкцій є «Вікна Steko». Аналізуючи цей сайт, можна відзначити декілька ключових аспектів, що стосуються його функціоналу, каталогу та калькулятора вартості.

Головна сторінка сайту містить основну інформацію про компанію, її продукти та послуги. Навігаційне меню надає доступ до різних розділів, таких як продукція, калькулятор вартості, новини та контакти. На головній сторінці також розміщені акції, відгуки клієнтів та інформація про переваги компанії.

Каталог продукції є одним з основних розділів сайту і поділений на кілька категорій, що включають вікна, двері, ролетні системи, підвіконня та аксесуари. Кожен продукт має детальний опис, технічні характеристики, фото та іноді відеоогляд. Також доступна інформація про енергозбереження, звукоізоляцію та інші властивості продукції.

Однією з важливих функцій сайту є калькулятор вартості, який дозволяє потенційним клієнтам розрахувати вартість необхідних конструкцій. Калькулятор зображений на рис 1.3.2

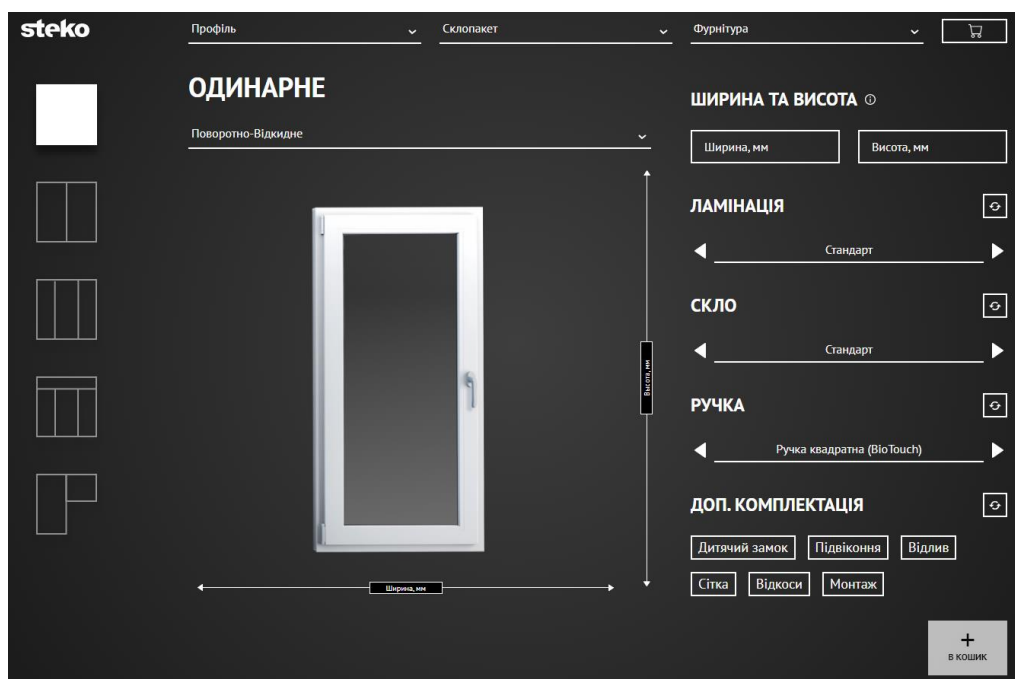


Рисунок 1.3.2 – Калькулятор компанії «Steko»

Калькулятор має зручний інтерфейс, що включає кілька етапів: вибір типу конструкції, налаштування параметрів (розміри, тип профілю, кількість камер у склопакеті, колір та інші характеристики), вибір додаткових опцій (москітні сітки, підвіконня, монтажні роботи) та розрахунок вартості. Після введення всіх параметрів калькулятор відображає приблизну вартість конструкції.

Серед інших функціональних можливостей сайту можна виділити онлайн-консультації, новини та акції, а також контакти та інформацію про партнерство. Консультації надаються через чат, що дозволяє клієнтам оперативно отримати відповіді на запитання. У розділі новин регулярно оновлюється інформація про діяльність компанії, нові продукти та спеціальні пропозиції. Контактна інформація включає адреси офісів та дилерів, а також можливості співпраці.

Перевагами сайту є його зручність, широкий асортимент продукції та детальна інформація про кожен товар. Калькулятор вартості значно полегшує

процес замовлення, дозволяючи клієнтам самостійно розрахувати вартість необхідних конструкцій. Онлайн-консультації та регулярні оновлення новин забезпечують високий рівень обслуговування клієнтів.

Недоліками можна вважати можливу складність для деяких користувачів при налаштуванні параметрів у калькуляторі вартості та необхідність додаткових консультацій для уточнення деяких деталей замовлення.

Загалом, сайт Steko.ua є ефективним інструментом для залучення та обслуговування клієнтів, надаючи їм всю необхідну інформацію для вибору та замовлення металопластикових конструкцій.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проєктування варіантів використання системи

У системі для розрахунку вартості віконних систем задіяно два основні актори: користувач і адміністратор. Кожен з цих акторів виконує специфічні функції та має чітко визначені ролі у рамках системи. Користувач відповідає за взаємодію з інтерфейсом системи для отримання необхідної інформації та проведення розрахунків, тоді як адміністратор здійснює управління контентом та підтримку системи, забезпечуючи її стабільне та ефективне функціонування. Актори зображені на рис. 2.1.1



Рисунок 2.1.1 – Актори застосування

В таблиці 2.1.1 наведені основні варіанти використання користувача в системі.

Таблиця 2.1.1 – Варіанти використання користувача

Актор	Дія	Опис дії
Користувач	Реєстрація	Користувач створює обліковий запис, вводячи необхідні дані, такі як ім'я, електронна адреса та пароль.
Користувач	Авторизація	Після реєстрації користувач входить в свій обліковий запис, вводячи валідний пароль та логін.

Продовження таблиці 2.1.1 – Варіанти використання користувача

Користувач	Перегляд інформації про продукцію	Користувач переглядає детальну інформацію про різні віконні системи, що доступні на сайті, включаючи характеристики та переваги кожного продукту.
Користувач	Відправка та заповнення заявки на консультацію	Користувач відправляє запит на консультацію, заповнивши форму з контактною інформацією та питаннями, для отримання додаткової інформації або рекомендацій.
Користувач	Вибір параметрів метлопластикових конструкцій	Користувач налаштовує параметри вікон відповідно до своїх потреб і вимог, вибираючи тип конструкції, матеріали та додаткові опції.
Користувач	Використання калькулятора вартості	Користувач вводить параметри металопластикової конструкції, такі як тип вікна, розміри, матеріал рами, для отримання розрахунку вартості.
Користувач	Фільтрація товарів за категоріями	Користувач застосовує фільтри в каталозі для сортування товарів за категоріями.
Користувач	Додавання товару в кошик	Користувач додає вибрані товари в кошик, готуючи їх до оформлення замовлення або подальшого редагування списку покупок.

В таблиці 2.1.2 наведені основні варіанти використання адміністратора.

Таблиця 2.1.2 – Варіанти використання адміністратора.

Актор	Дія	Опис дії
Адміністратор	Керування користувачами	Адміністратор переглядає, видаляє, або отримує список всіх користувачів.
Адміністратор	Додавання та редагування продуктів	Адміністратор може додавати нові продукти до каталогу, редагувати їхні характеристики, ціни та описи, а також видаляти застарілі або непотрібні позиції.
Адміністратор	Обробка заявок на консультацію	Адміністратор отримує запити на консультацію від користувачів, переглядає їх, відповідає на них або перенаправляє до відповідних фахівців для консультації.

За допомогою отриманих даних з таблиць про основні можливості та дії користувачів і адміністратора, ми можемо перейти до побудови діаграми варіантів використання [3]. Це зображено на рис 2.1.2 та 2.1.3

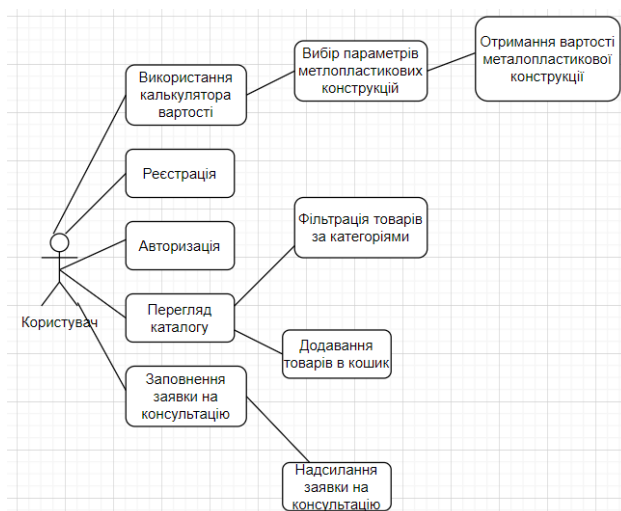


Рисунок 2.1.2 – Варіанти використання користувача

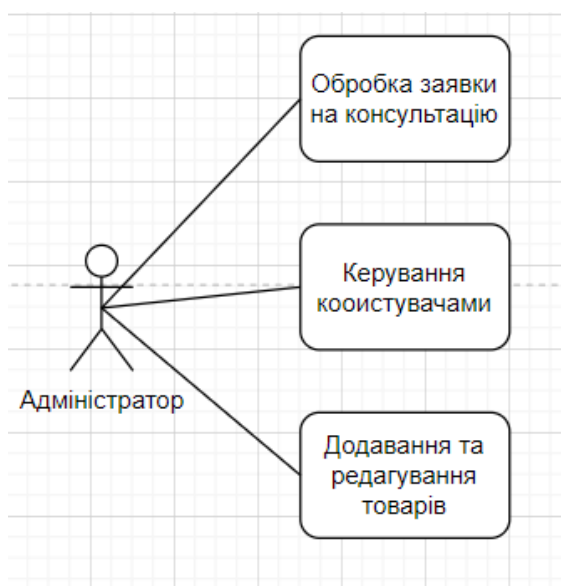


Рисунок 2.1.3 – Варіанти використання адміністратора

2.2 Проектування інформаційної системи

Діаграми послідовності є важливим інструментом для моделювання послідовності взаємодій між об'єктами в системі [4]. По-перше, вона забезпечує чітке уявлення про процес взаємодії між різними компонентами системи. Це допомагає розробникам і проєктувальникам зрозуміти, як відбувається обмін повідомленнями між об'єктами на кожному етапі розрахунку вартості металопластикових конструкцій. По-друге, діаграма послідовності сприяє ефективній комунікації між членами команди, надаючи візуальне представлення процесів, що спрощує обговорення та виявлення можливих проблем.

Подана діаграма показує процес авторизації користувача на веб-сайті. Користувач вводить свій логін та пароль у форму на веб-сайті. Система отримує введені користувачем дані і шукає користувача в базі даних за його логіном. База даних повертає відповідь системі про наявність користувача з таким логіном. Система перевіряє, чи введений користувачем пароль збігається з паролем, збереженим в базі даних для цього користувача. Якщо пароль введено правильно і

користувач існує в базі даних, система виконує авторизацію користувача. Після цього система повідомляє користувача про успішну або неуспішну авторизацію. Це зображено на рисунку 2.2.1

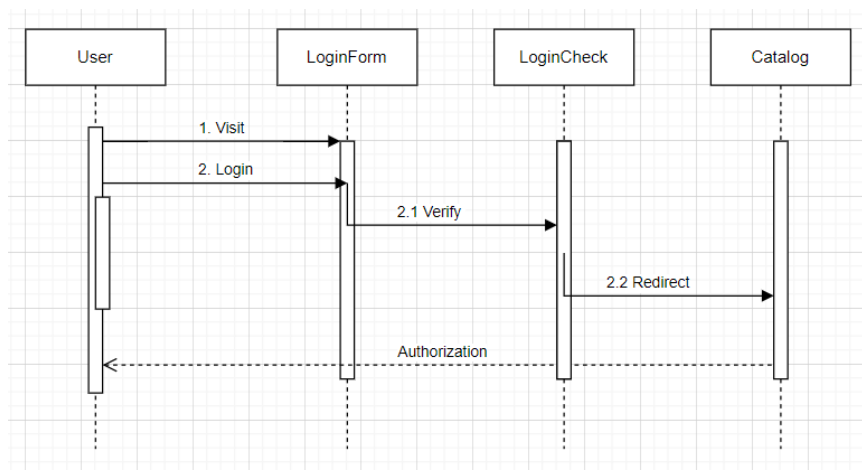


Рисунок 2.2.1 – Діаграма послідовності авторизації

Процес фільтрації товарів за категоріями. Користувач взаємодіє з інтерфейсом веб-сайту і обирає категорію товарів або встановлює параметри фільтрації. Після вибору категорії або параметрів фільтрації, система отримує ці дані і звертається до бази даних для отримання відповідного списку товарів. База даних, отримавши запит від системи, проводить фільтрацію товарів відповідно до заданих критеріїв. Після цього вона повертає відфільтрований список товарів системі. Система отримує від бази даних список товарів, що відповідають вибраним критеріям, і відображає цей список користувачеві на веб-сайті. Це зображено на рисунку 2.2.2

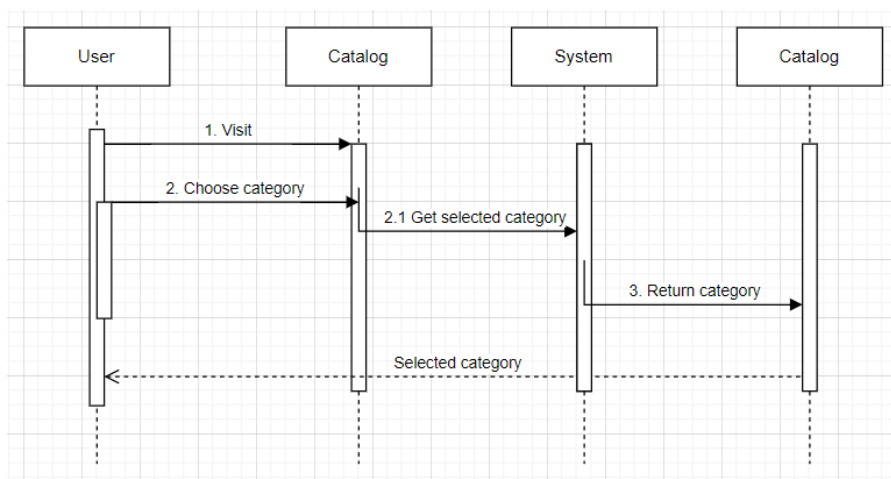


Рисунок 2.2.2 – Діаграма послідовності фільтрації

Процес запиту на консультацію. Користувач натискає кнопку "Розрахувати вартість" на веб-сайті. Після цього система відображає форму, де користувач вводить своє ім'я, контактні дані та вказує тип і розмір конструкції. Після введення даних сторінка перевіряє їх правильність та відправляє серверу. Після успішної перевірки сервер автоматично надсилає повідомлення адміністратору про новий запит на консультацію. Після цього система надсилає користувачеві підтвердження успішного надсилання запиту на консультацію. Це зображено на рис 2.2.3

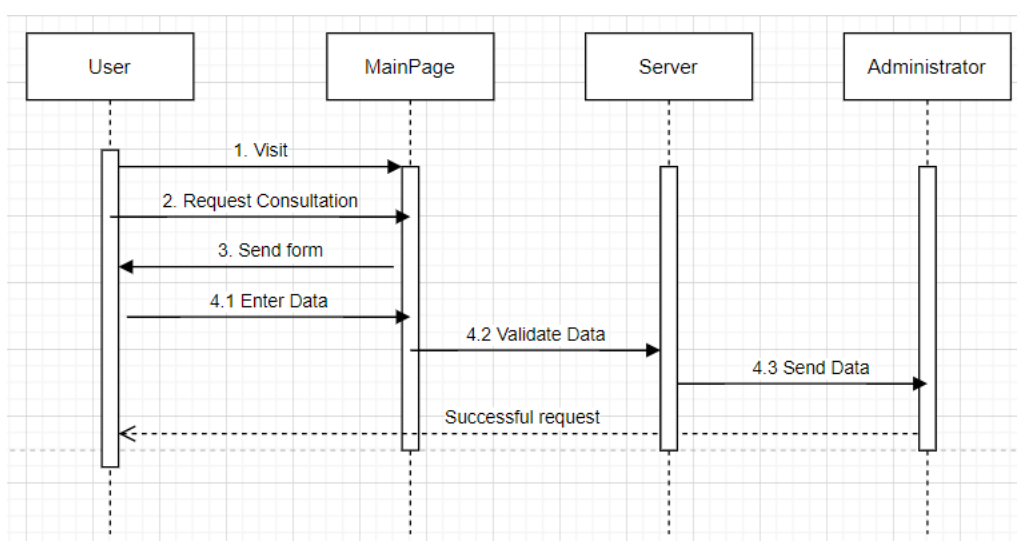


Рисунок 2.2.3 – Діаграма послідовності запиту на консультацію

Діаграми послідовності допомогли краще зрозуміти і моделювати взаємодії між різними компонентами системи, що дозволило виявити та виправити потенційні проблеми на ранніх етапах розробки. Крім того, вони допомогли ефективніше управляти складними процесами та забезпечили основу для успішної реалізації проєкту.

2.3 Проєктування бази даних

У системі для обрахунку вартості одним з важливих аспектів є підключення бази даних. База даних забезпечує надійне зберігання та ефективне управління даними, необхідними для функціонування веб-додатку. В даному проєкті використовується MongoDB — документо-орієнтована система управління базами даних, яка дозволяє зберігати дані у вигляді документів формату JSON [5].

Підключення бази даних до проєкту є критично важливим аспектом розробки будь-якого сучасного програмного забезпечення. Це дозволяє зберігати, організовувати та обробляти великі обсяги інформації ефективно і безпечно. Завдяки інтеграції з базою даних, програма може забезпечувати швидкий доступ до необхідних даних, забезпечувати їхню цілісність та підтримувати синхронізацію між різними компонентами системи.

Загальний вид бази даних можна представити як сукупність колекцій, кожна з яких відповідає за певний тип даних. Кожна колекція містить документи, що зберігають інформацію у структурованому вигляді. Структура бази даних зображена на рис. 2.3.1

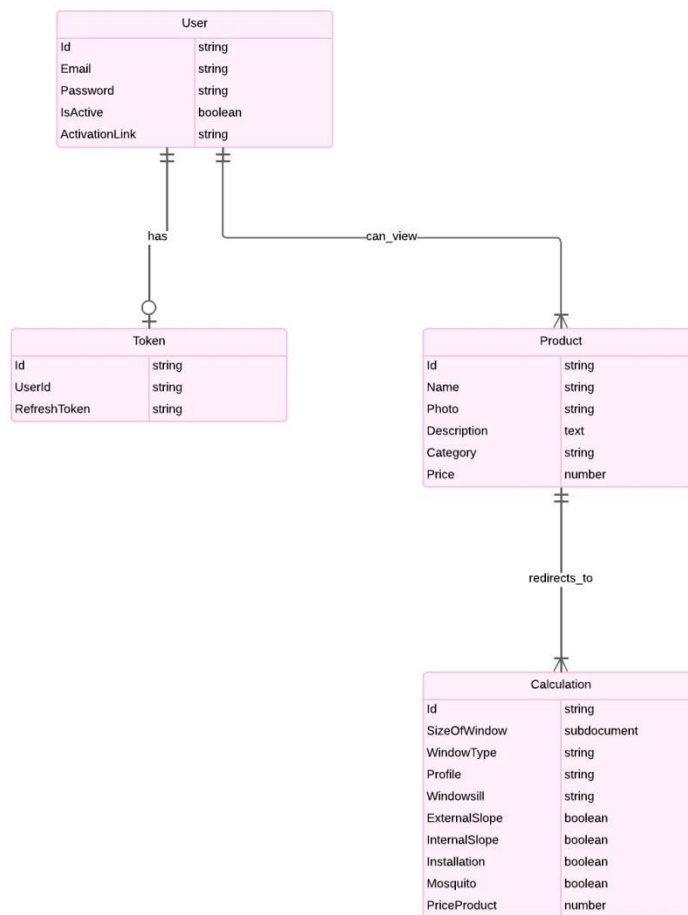


Рисунок 2.3.1 – Структура бази даних

Колекція користувачів містить інформацію про зареєстрованих користувачів додатку. Кожен документ у цій колекції містить унікальний ідентифікатор користувача, електронну пошту, захешований пароль, прапорець чи активований акаунт та лінк для активації облікового запису. Опис полів в колекції для користувача наведені в таблиці 2.3.1

Таблиця 2.3.1 Опис колекції користувача

Назва	Тип	Опис	Обмеження	Значення за замовчуванням
_id	String	Унікальний ідентифікатор користувача	Унікальне, обов'язкове	—

Продовження таблиці 2.3.1 – Опис колекції користувача

email	String	Електронна пошта	Унікальне, обов'язкове, нижній регістр, має містити «@»	—
password	String	Захешований пароль	Обов'язкове, мінімальна довжина 5 символів	—
isActive	Boolean	Чи активований акаунт	Обов'язкове	false
activationLink	String	Посилання для активації акаунту	Обов'язкове	—

Приклад зберігання користувача в колекції зображено на рис. 2.3.2

```

_id: ObjectId('666031bba6e1e23ba763038d')
email: "nazar@gmail.com"
password: "$2b$04$emTYcCPZ7aMFBd4jI3ye/eVRhR8fVILbLlvhoXEz578M2/YIji fAS"
isActive: false
activationLink: "5932e67a-4926-44b7-86c2-8ac42d631173"

```

Рисунок 2.3.2 – Приклад зберігання користувача

Колекція токенів містить інформацію про рефреш-токени, які використовуються для авторизації користувачів у додатку. Кожне поле у цій колекції містить унікальний ідентифікатор документа, ідентифікатор користувача, до якого належить токен, та сам рефреш-токен. Опис полів в колекції для токенів користувача наведені в таблиці 2.3.2

Таблиця 2.3.2 Опис колекції токенів користувача

Назва	Тип	Опис	Обмеження	Значення за замовчуванням
_id	String	Унікальний ідентифікатор токена	Унікальне, обов'язкове	—
user	String	Ідентифікатор користувача	Обов'язкове, посилання на користувача	—
refreshToken	String	Рефреш-токен	Обов'язкове	—

Приклад зберігання токенів користувача в колекції зображено на рис. 2.3.3

```
_id: ObjectId('666049faa6e1e23ba76303a0')
user : ObjectId('666031bba6e1e23ba763038d')
refreshToken : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlZ3MTc3Njc2MTAsImV4cCI6Mm..."
```

Рисунок 2.3.3 – Приклад зберігання токенів користувача

Колекція товарів містить інформацію про кожен товар, доступний у магазині. Кожен документ у цій колекції містить унікальний ідентифікатор поля, унікальний ідентифікатор товару, назву, зображення, категорію та ціну. Опис полів в колекції для токенів користувача наведені в таблиці 2.3.3

Таблиця 2.3.3 Опис колекції товарів каталогу

Назва	Тип	Опис	Обмеження	Значення за замовчуванням
id	Number	Унікальний ідентифікатор	Унікальне, обов'язкове	—
title	String	Назва товару	Обов'язкове	—

Продовження таблиці 2.3.3 – Опис колекції товарів каталогу

img	String	Назва зображення	Обов'язкове	—
category	String	Категорія товару	Обов'язкове	—
price	Number	Ціна товару	Обов'язкове	—

Приклад зберігання товарів каталогу в колекції зображено на рис. 2.3.4

```

id : 7
title : "Алюмінієве вікно"
img : "aluminium_3.jpg"
category : "aluminium"
price : "4799"

```

Рисунок 2.3.4 – Приклад зберігання товарів каталогу

Колекція результатів обчислення вартості містить інформацію про кожну конструкцію, розраховану за допомогою калькулятора в додатку. Кожен документ у цій колекції містить унікальний ідентифікатор документа, розміри вікна(висота та ширина), тип вікна, профіль, підвіконня, зовнішній та внутрішній відкоси, монтаж, москітну сітку та ціну продукту. Опис полів в колекції для результатів обчислення вартості наведені в таблиці 2.3.4

Таблиця 2.3.4 Опис колекції результатів обчислення вартості

Назва	Тип	Опис	Обмеження	Значення за замовчуванням
_id	String	Унікальний ідентифікатор документа	Унікальне, обов'язкове	—
sizeOfWindow	Object	Розміри вікна	Обов'язкове	—

Продовження таблиці 2.3.4 – Опис колекції результатів обчислення вартості

windowType	String	Тип вікна	Обов'язкове	—
profile	String	Профіль вікна	Обов'язкове	—
windowsill	String	Назва підвіконня	Обов'язкове	—
externalslope	Boolean	Зовнішній відкос	Обов'язкове	false
internalslope	Boolean	Внутрішній відкос	Обов'язкове	false
installation	Boolean	Монтаж	Обов'язкове	false
mosquito	Boolean	Москітна сітка	Обов'язкове	false
priceProduct	Number	Ціна продукту	Обов'язкове	—

Приклад зберігання результатів обчислення вартості в колекції зображено на рис. 2.3.5

```

_id: ObjectId('6669bb3bdda0875cadd4fc64')
▼ sizeOfWindow: Object
  width: 963
  height: 1366
windowType: "big_french"
profile: "standart"
windowsill: "brusbox-550"
externalslope: true
internalslope: false
installation: true
mosquito: "roll"
priceProduct: 6849

```

Рисунок 2.3.5 – Приклад зберігання результатів обчислення вартості

Ці колекції разом утворюють структуру бази даних додатку, забезпечуючи організоване зберігання даних про товари, користувачів, токени авторизації та розрахунки калькулятора.

3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір основних технологій веб-розробки

Визначення з основними технологіями веб-розробки є критично важливим етапом, який визначає її ефективність, продуктивність і можливості для подальшого розвитку. Для даного проєкту, що передбачає створення системи обрахунку вартості металопластикових конструкцій, необхідно обрати технології, які забезпечать швидку та надійну обробку даних, зручний інтерфейс користувача та можливість масштабування.

Однією з основних технологій, яку було обрано для даного проєкту, є JavaScript [6]. Це мова програмування, що дозволяє створювати інтерактивні веб-сторінки і додатки.

Основні причини вибору JavaScript для обраного сайту включають:

- Динамічність та інтерактивність: Мова дозволяє створювати динамічні та інтерактивні веб-сторінки. Для системи, яка включає складний калькулятор вартості, важливо забезпечити користувачам можливість швидко і зручно вводити дані та отримувати результати. JavaScript дозволяє реалізувати динамічні елементи, такі як форми, кнопки, таблиці та графіки, які оновлюються без перезавантаження сторінки.
- Широка підтримка браузером: Мова підтримується всіма сучасними веб-браузерами, що гарантує коректне відображення і функціонування нашого сайту на різних пристроях і платформах. Це дозволяє охопити широку аудиторію користувачів незалежно від того, який браузер вони використовують.
- Фреймворки та бібліотеки: JavaScript має багатий набір фреймворків і бібліотек, які значно спрощують розробку і прискорюють процес створення веб-додатків. Для нашого проєкту ми використаємо такі популярні фреймворки, як React, Node.js, які забезпечують ефективну організацію коду і покращують продуктивність. Крім того, бібліотеки, такі як jQuery, можуть спростити маніпуляції з DOM і обробку подій.

- Односторінкові додатки: Використовуючи JavaScript, можливо розробити односторінковий додаток, де вся взаємодія з сервером відбувається через AJAX-запити або використання технології Fetch API. Це дозволяє забезпечити швидке завантаження сторінок і плавний користувацький досвід, що є важливим для нашого калькулятора вартості.

- Реалізація логіки на клієнтській стороні: JavaScript дозволяє виконувати частину логіки на клієнтській стороні, що зменшує навантаження на сервер і підвищує загальну продуктивність системи. Наприклад, валідацію введених даних можна здійснювати безпосередньо в браузері користувача, що дозволяє оперативно виявляти помилки і підвищує зручність використання.

React – це популярна бібліотека, яка призначена для створення користувацьких інтерфейсів. Вона дозволяє розробникам створювати масштабовані і продуктивні веб-додатки за допомогою компонентного підходу. У нашому проєкті, присвяченому створенню системи обрахунку вартості металопластикових конструкцій, React відіграє ключову роль у побудові зручного і динамічного інтерфейсу користувача а також калькулятора вартості метало пластикових конструкцій [7].

- Компонентний підхід: React дозволяє розробляти інтерфейс користувача за допомогою компонентів, які можуть бути повторно використані в різних частинах програми. Кожен компонент є незалежним і відповідає за свою частину інтерфейсу. Це спрощує розробку, тестування і підтримку коду, оскільки зміни в одному компоненті не впливають на інші компоненти.

- Віртуальний DOM: Однією з основних переваг React є використання віртуального DOM. Віртуальний DOM є абстракцією реального DOM, що дозволяє React ефективно оновлювати і рендерити компоненти. Коли стан або пропси компонента змінюються, React створює новий віртуальний DOM і порівнює його з попереднім станом, вносячи тільки необхідні зміни в реальний DOM. Це значно підвищує продуктивність і швидкість роботи додатку.

- Одностороння передача даних: React використовує односторонню передачу даних, що означає, що дані завжди передаються від батьківського

компонента до дочірнього через пропси. Це забезпечує передбачуваність і контрольованість стану додатку, що полегшує налагодження і відстеження змін.

- **JSX:** React використовує JSX – розширення синтаксису JavaScript, що дозволяє писати HTML-подібний код всередині JavaScript. JSX спрощує створення компонентів і надає можливість використовувати повний функціонал JavaScript при розробці інтерфейсу.

Node.js – це середовище, яке дозволяє запускати JavaScript-код на сервері.

Переваги використання Node.js:

- Node.js використовує неблокуючу, асинхронну модель вводу/виводу, що дозволяє обробляти велику кількість одночасних запитів з мінімальними витратами ресурсів. Це забезпечує високу продуктивність і швидкість роботи нашої системи.

- За допомогою Node.js можна легко інтегруватись з базою даних MongoDB для зберігання і отримання необхідної інформації.

- Node.js буде використовуватися для створення API, яке забезпечить взаємодію між фронтендом і бекендом. Це дозволить фронтенд-додатку, розробленому на React, надсилати запити до сервера і отримувати необхідні дані для розрахунків.

JWT – це відкритий стандарт для безпечної передачі інформації між сторонами. Ця технологія є ефективним способом автентифікації та авторизації користувачів у веб-додатках [8]. У нашому проєкті з розробки інформаційної системи для обрахунку вартості металопластикових конструкцій використання JWT допоможе забезпечити безпеку та зручність роботи з користувачами.

Переваги використання JWT:

Безпека: JWT забезпечує високий рівень безпеки завдяки використанню криптографічних методів підпису. Токени можуть бути підписані за допомогою секретного ключа або асиметричних ключів, що гарантує цілісність і автентичність даних.

Самодостатність: Кожен JWT містить всю необхідну інформацію для автентифікації, включаючи дані про користувача і термін дії токена. Це дозволяє серверам легко перевіряти автентичність запитів без необхідності зберігання стану сеансу або пошуку даних у базі даних.

Зручність використання: JWT є компактними і можуть бути легко передані через URL, параметри запиту або заголовки HTTP. Це робить їх ідеальними для використання в сучасних веб-додатках, де вони можуть бути використані для автентифікації API-запитів і захисту ресурсів.

Bootstrap – це найпопулярніший фреймворк для розробки адаптивних веб-додатків. Він містить набір інструментів для створення інтерфейсів користувача, включаючи готові компоненти і стилі CSS, а також JavaScript плагіни[7]. У нашому проєкті, присвяченому створенню системи обрахунку вартості металопластикових конструкцій, використання Bootstrap допоможе швидко і ефективно створити зручний і привабливий інтерфейс користувача.

- Адаптивний дизайн: Bootstrap забезпечує створення адаптивних веб-сторінок, які добре виглядають на різних пристроях і розмірах екранів. Це дозволяє нам розробити інтерфейс, який буде однаково зручним для користувачів на комп'ютерах, планшетах і смартфонах.

- Швидка розробка: Використання готових компонентів і шаблонів Bootstrap значно прискорює процес розробки інтерфейсу. Розробникам не потрібно витрачати час на створення базових стилів і компонентів з нуля, що дозволяє зосередитися на унікальних аспектах нашого додатку.

- Єдиний стиль: Bootstrap забезпечує єдиний стиль і дизайн для всіх компонентів додатку, що створює гармонійний і професійний вигляд інтерфейсу. Це також полегшує підтримку коду, оскільки всі стилі організовані і структуровані в єдиному фреймворку.

- Гнучкість і налаштування: Bootstrap дозволяє легко налаштовувати стилі і компоненти відповідно до вимог проєкту. Можна змінювати кольорову схему, типографіку, розміри елементів та інші параметри, щоб створити унікальний вигляд інтерфейсу.

3.2 Вибір середовища розробки

У процесі розробки програмного забезпечення правильний вибір середовища розробки є ключовим фактором, який може значно вплинути на ефективність та зручність роботи розробників. Для розробки нашого веб-сайту було обрано середовище розробки Visual Studio Code. Це рішення обґрунтоване низкою вагомих причин, які будуть детально розглянуті нижче.

Основні характеристики Visual Studio Code:

- Підтримка різних мов програмування: Visual Studio Code підтримує величезну кількість мов програмування, включаючи HTML, CSS, JavaScript, TypeScript, Python, C++, Java та багато інших. Це дозволяє розробникам використовувати одне середовище для роботи над проектами, написаними різними мовами, що значно спрощує процес розробки та знижує необхідність переключатися між різними редакторами.
- Розширюваність: VS Code має вбудовану підтримку розширень, які можна встановлювати для додавання нових функцій. Існує величезна кількість розширень, які можна знайти у Marketplace, включаючи інструменти для дебагінгу, інтеграції з системами контролю версій, підтримку різних фреймворків та бібліотек, а також теми оформлення та значки. Це дозволяє налаштовувати середовище розробки під конкретні потреби проекту та вподобання розробника.
- Інтелектуальні функції редагування коду: VS Code забезпечує інтелектуальні функції автозавершення коду, підказки, рефакторинг та навігацію по коду. Це значно підвищує продуктивність розробника, дозволяючи швидко та безпомилково писати код. Такі функції як автозавершення синтаксису та рефакторинг допомагають зменшити кількість помилок та покращити якість коду.
- Вбудований дебагер: Visual Studio Code містить вбудований дебагер, який підтримує дебагінг JavaScript, TypeScript, Python та інших мов. Це дозволяє

відстежувати та виправляти помилки в коді безпосередньо у середовищі розробки, що значно спрощує процес тестування та налагодження.

- Кросплатформеність: Visual Studio Code є кросплатформенним середовищем, що дозволяє розробникам працювати на будь-якій операційній системі. Це особливо важливо для команд, які використовують різні операційні системи, оскільки забезпечує єдине середовище розробки для всіх членів команди.

Для розробки нашого веб-сайту вибір Visual Studio Code був зумовлений кількома ключовими факторами. По-перше, підтримка різних мов програмування, включаючи HTML, CSS, JavaScript, дозволяє використовувати одне середовище для всього процесу розробки. Це зручно і підвищує продуктивність розробників.

По-друге, можливість розширення функціоналу через встановлення різноманітних розширень дозволяє адаптувати VS Code під конкретні потреби проєкту. Наприклад, розширення для React та Node.js забезпечують інструменти для більш зручної роботи з цими технологіями, а інструменти для дебагінгу і тестування допомагають швидко виявляти та виправляти помилки.

Загалом, Visual Studio Code є потужним та гнучким середовищем розробки, яке забезпечує всі необхідні інструменти для створення сучасних веб-додатків. Його можливості дозволяють швидко та ефективно розробляти, тестувати та налагоджувати код, що робить його ідеальним вибором для нашого проєкту.

3.3 Реалізація веб-застосунку

Головна сторінка веб-сайту є важливим елементом, що забезпечує зручний та інтуїтивно зрозумілий доступ до всіх функцій та розділів. Нижче надається детальний опис структури та функціональних можливостей головного вікна.

У верхній частині головного вікна розташований логотип компанії з підписом "Встановлення металопластикових конструкцій". Логотип слугує не

тільки декоративним елементом, але й допомагає користувачам швидко повернутися на головну сторінку при кліку на нього.

Головне меню представлено у вигляді кнопки "Увійти", яка відкриває нове вікно для авторизації користувача. Це меню забезпечує швидкий доступ до інших розділів, які потребують авторизації.

Інформаційні блоки містять важливі дані про робочі години компанії, контактну інформацію та можливість замовити зворотний дзвінок. Вони допомагають користувачам швидко знайти необхідну інформацію та зв'язатися з представниками компанії.

Центральний банер містить інформацію про акційні пропозиції та ключові послуги компанії. Це привертає увагу користувачів та стимулює їх до дії. Під банером розташовані блоки з ключовими перевагами послуг компанії: висока якість, швидкий монтаж, гарантія 3 роки та вивіз сміття. Ці блоки допомагають користувачам швидко ознайомитися з основними перевагами пропозицій компанії.

У правій частині головного вікна розташована форма для замовлення безкоштовного заміру. Користувачі можуть заповнити своє ім'я та телефон, щоб викликати замірника. Це зображено на рис. 3.3.1

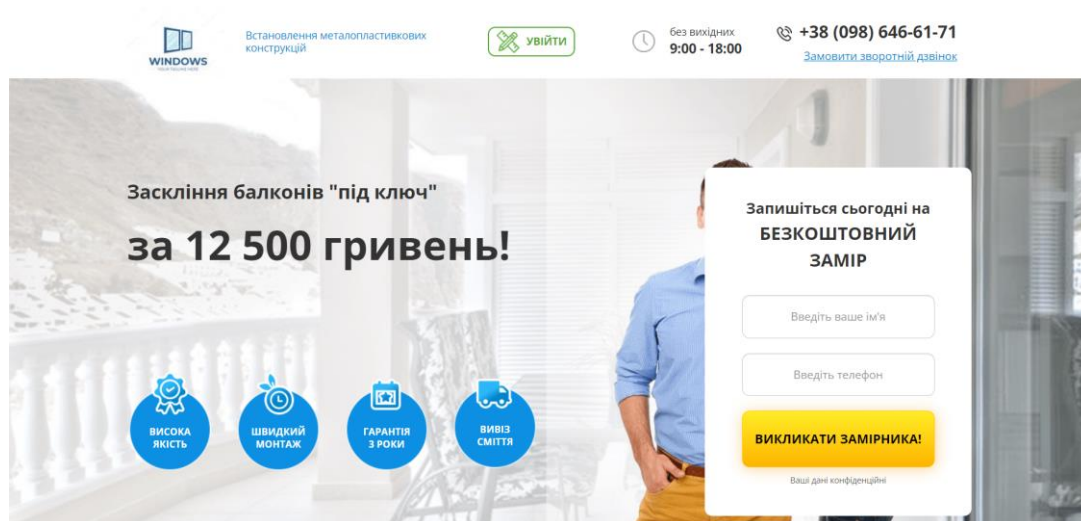


Рисунок 3.3.1 – Головна сторінка веб-сайту

Секція "Заскління балконів та лоджій" є важливою частиною головної сторінки нашого веб-сайту, яка надає користувачам детальну інформацію про різні типи заскління та їх особливості. Нижче надається детальний опис структури та функціональних можливостей цієї секції.

У верхній частині секції розташований заголовок "Заскління балконів та лоджій", що привертає увагу користувачів та інформує їх про зміст секції.

Під заголовком розташований слайдер, який містить різні типи заскління, такі як дерев'яне, алюмінієве, пластикове, французьке (панорамне) та заскління з виносом. Кожен варіант представлений у вигляді блока з іконкою та назвою, що дозволяє користувачам швидко ознайомитися з доступними опціями. Це зображено на рис. 3.3.2

ЗАСКЛІННЯ БАЛКОНІВ ТА ЛОДЖІЙ



[Дерев'яне
скління](#)



[Алюмінієве
скління](#)



[Заскління
пластиковими
рамами](#)



[Французьке
скління
\(панорамне\)](#)



[Скління
з виносом](#)

ХОЛОДНЕ



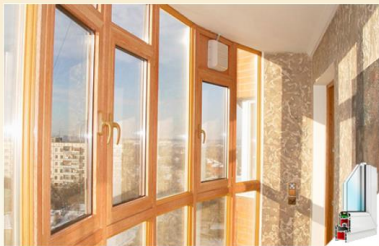
- ✓ Конструктивна товщина 42-58 мм
- ✓ Скло: поліроване скло (товщиною 4 - 5 мм)
- ✓ Теплоізоляція: 0,07 м² * С/Вт
- ✓ Звукоізоляція: 20-27 дБ

2600 грн.кв.м.

під ключ з встановленням

РОЗРАХУВАТИ ВАРТІСТЬ

ТЕПЛЕ



- ✓ Конструктивна товщина профіля: 58-78 мм
- ✓ Скло: однокамерні и двокамерні склопакети
- ✓ Теплоізоляція: 0,63 - 1,05 м² * С/Вт
- ✓ Звукоізоляція: до 5 класу

4000 грн.кв.м.

під ключ з встановленням

РОЗРАХУВАТИ ВАРТІСТЬ

Рисунок 3.3.2 – Секція "Заскління балконів та лоджій"

Секція "Оздоблення балкону" пропонує комплексні послуги з внутрішнього та зовнішнього оздоблення балконів, а також виносне скління та встановлення даху. Вона оформлена з яскравими зображеннями та детальними описами матеріалів, що використовуються, а також включає інтерактивний елемент – форму для запису на безкоштовний замір. Кожна підкатегорія оздоблення представлена окремим блоком з відповідними цінами та зображеннями матеріалів, що робить секцію інформативною та зручною для користувача.

Секція містить слайдер, де відвідувач може обрати одну з чотирьох категорій оздоблення: внутрішнє оздоблення, зовнішнє оздоблення, виносне скління та дах на балкон. Кожна категорія представлена окремим блоком з назвою та зображенням.

Після вибору категорії користувачеві відкривається відповідний блок з детальною інформацією про різні види матеріалів для оздоблення балкона, їх ціни та зображення.

Кожна категорія оздоблення містить візуальні зображення матеріалів, що допомагає відвідувачам уявити, як буде виглядати їхній балкон після оздоблення. Ціни на матеріали вказані за квадратний метр, що дозволяє легко оцінити витрати на оздоблення. Це зображено на рис. 3.3.3

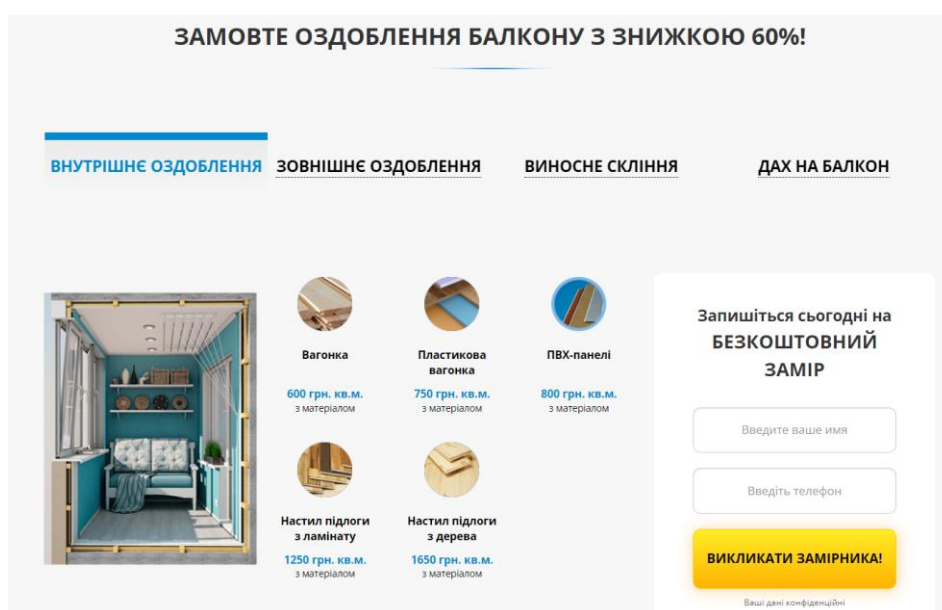


Рисунок 3.3.3 – Секція "Оздоблення балкону"

Секція "Наші роботи" призначена для представлення портфоліо робіт компанії. Ця секція використовується для візуального демонстрування та висвітлення спеціалізованої діяльності

Весь вміст розташований у контейнері, що забезпечує зручну організацію і розміщення елементів. Секція включає рядок з зображеннями, які відображають попередній перегляд портфоліо компанії.

Кожне мініатюрне зображення має посилання на велике зображення роботи, яке відкривається у новому вікні або вікні легкого боксу (lightbox).

Секція має адаптивний дизайн, що забезпечує коректне відображення на різних пристроях. Це досягається за допомогою використання класів, що адаптують розміщення блоків для різних розмірів екрану. Це зображено на рис. 3.3.4

НАШІ РОБОТИ



Рисунок 3.3.4 – Секція "Наші роботи"

Наступна секція призначена для наголошення на основних гарантіях і перевагах, які компанія пропонує своїм клієнтам. Вона має високий ступінь візуальної привабливості та чіткість у викладенні інформації.

Весь контент розміщений у контейнері, що дозволяє зручно організувати та розмістити елементи на сторінці. Кожна гарантія представлена у вигляді блоку,

який включає зображення та заголовок. Кожний блок з'являється з ефектом анімації, що надає динамічності і привертає увагу користувача.



Рисунок 3.3.4 – Секція "Гарантії"

Секція "Таймер" призначена для просування акційної пропозиції компанії, спрямованої на заощадження клієнтів при покупці послуг або товарів. Вона включає в себе інформацію про саму акцію, термін її дії, а також форму для запису на безкоштовний замір.

Під підзаголовком розміщений таймер, що підраховує час який залишився до завершення акції. Він візуалізує час залишку у форматі "дні - години - хвилини - секунди", що додає почуття невідкладності. Справа від таймера розташована форма. Вона містить поля для введення імені та телефону користувача, а також кнопку для надсилання форми. Це зображено на рис. 3.3.5

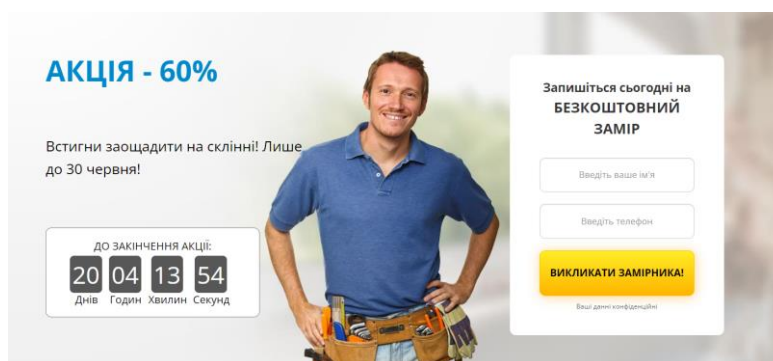


Рисунок 3.3.5 – Секція "Таймер"

Секція "Як нас знайти" призначена для надання відвідувачам інформації про контактні дані компанії, включаючи адреси, телефон, електронну пошту та режим роботи.

Ліворуч від текстової інформації розташована карта Google Maps, яка відображає місцезнаходження компанії. Карта є інтерактивною та дозволяє відвідувачам швидко зорієнтуватися в розташуванні.

Проаворуч від карти знаходиться інформація, яка надає відвідувачам можливість зв'язатися з компанією за різними каналами зв'язку, включаючи телефон та електронну пошту, а також дає розуміння часу, коли компанія доступна для обслуговування. Це зображено на рис. 3.3.6

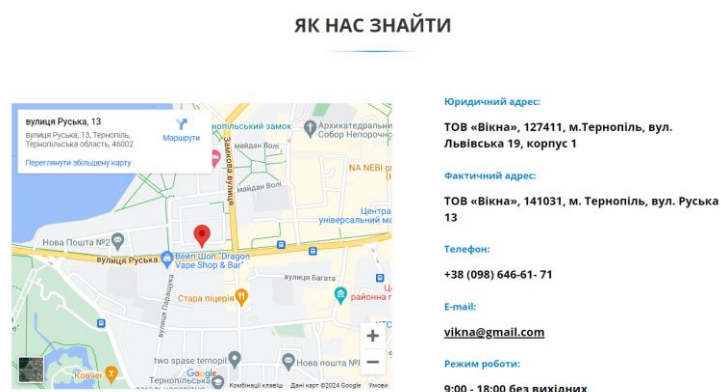


Рисунок 3.3.6 – Секція "Як нас знайти"

На веб-сайті існує секція, яка дозволяє користувачам розраховувати вартість послуги заскління балконів за допомогою трьох модальних вікон.

Перше вікно призначене для вибору форми балкона та введення його розмірів. Користувач обирає одну з запропонованих іконок зображень балконів, після чого вводить ширину та висоту у відповідні поля введення. Після натискання кнопки "Далі", відкривається наступне модальне вікно.

В цьому вікні користувач обирає тип скління та його профіль. Після вибору параметрів скління і натискання кнопки "Далі", відкривається останнє модальне вікно.

Третє вікно призначене для введення контактних даних користувача та отримання приблизної вартості заскління балкона. Користувач вводить своє ім'я та номер телефону у відповідні поля введення. Після натискання кнопки "Розрахувати вартість", дані відправляються для обробки. Це зображено на рис 3.3.7

The figure displays three sequential screenshots of a web application interface for balcony glazing calculation.

- Скріншот 1 (ліворуч):** Вікно з заголовком "Калькулятор". Підзаголовок: "Виберіть форму балкона и вкажіть розміри". Показані іконки різних форм балконів. Нижче зображено вибраний варіант з трьома секціями. Під ним поля для вводу "Ширина мм" та "Висота мм". На дні жовта кнопка "Далі".
- Скріншот 2 (середина):** Вікно з заголовком "Калькулятор". Підзаголовок: "Виберіть тип скління та його профіль". Показано вибраний профіль "Алюмінієве скління" у випадаючому списку. Нижче є дві іконки: "Холодне" (сніжинка) з позначкою ☒ та "Тепле" (сонце) з позначкою ☐. На дні жовта кнопка "Далі".
- Скріншот 3 (праворуч):** Вікно з заголовком "Дякуємо за звернення! Залиште свої дані". Показані два поля для введення: "Введіть ваше ім'я" та "Введіть телефон". Нижче велика жовта кнопка "РОЗРАХУВАТИ ВАРТІСТЬ". Під нею дрібним шрифтом: "Передзвонимо протягом 10 хвилин".

Рисунок 3.3.7 – Модальні вікна

Система авторизації та реєстрації користувачів виконує важливу роль для забезпечення безпеки та зручності доступу до функціональності системи.

На сторінці авторизації користувачі можуть ввести свої облікові дані, такі як логін та пароль, для отримання доступу до системи. Основним елементом цієї сторінки є форма для входу, яка містить поля для введення логіну та пароля. Кнопка входу відправляє дані форми на сервер для перевірки автентифікації. У разі некоректного введення даних або інших помилок, система відображає відповідні повідомлення про помилки.

Сторінка реєстрації дозволяє новим користувачам створити обліковий запис. Форма для реєстрації містить поля для введення електронної пошти та пароля. Після заповнення форми користувач натискає кнопку реєстрації, яка відправляє дані на сервер для створення нового облікового запису. У разі помилок у введенні даних система відображає відповідні повідомлення, що дозволяє

користувачеві виправити їх. На сторінці реєстрації також є посилання, яке перенаправляє користувачів на сторінку авторизації, якщо у них вже є обліковий запис.

Для забезпечення безпеки даних використано валідацію як на стороні клієнта, так і на стороні сервера. Це включає перевірку правильності введених даних, таких як формат електронної пошти. Після успішної автентифікації користувач отримує JSON Web Token (JWT), який використовується для захисту подальших запитів до сервера. Використання JWT дозволяє забезпечити безпеку та зручність доступу до системи, оскільки токени можуть бути легко передані через заголовки HTTP.

Ще одним важливим аспектом є використання refresh tokenів, які дозволяють автоматично оновлювати access токени, забезпечуючи безперервний доступ користувачів до системи без необхідності повторної авторизації. Це підвищує зручність використання системи та знижує ризик втрати доступу через закінчення терміну дії токenu.

На сторінці реєстрації також реалізована функція активації облікового запису через електронну пошту. Після реєстрації користувач отримує лист з посиланням для активації свого облікового запису. Це забезпечує додатковий рівень безпеки, оскільки тільки користувачі, які мають доступ до вказаної електронної пошти, можуть активувати свої облікові записи[10].

Таким чином, сторінки авторизації та реєстрації в нашій системі забезпечують надійну, безпечну та зручну взаємодію користувачів з додатком, що є важливим для ефективного функціонування інформаційної системи обрахунку вартості металопластикових конструкцій.

Структура сторінки для авторизації та реєстрації зображена на рис.3.3.8

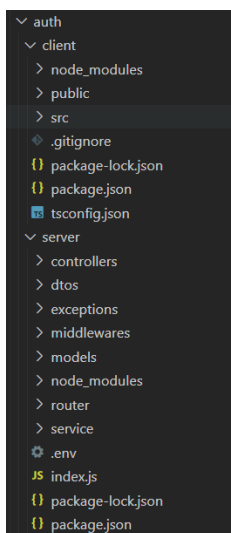


Рисунок 3.3.8 – Структура сторінки авторизації та реєстрації

На рисунку 3.3.9 зображений інтерфейс сторінки авторизації та реєстрації.

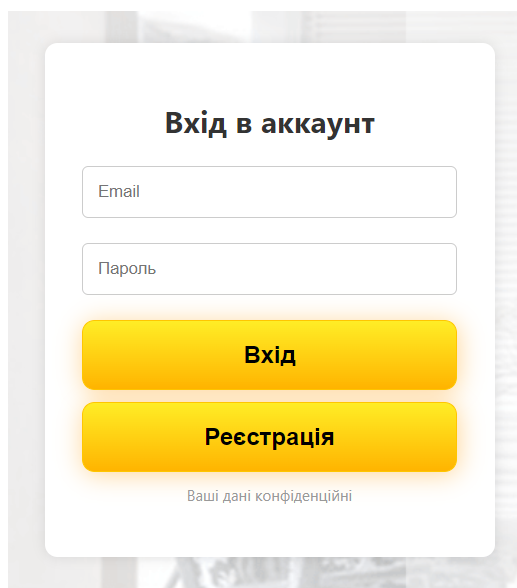


Рисунок 3.3.9 – Інтерфейс сторінки авторизації та реєстрації

Також до сторінки авторизації та реєстрації підключено базу даних MongoDB. На веб-сайті функціонал бази даних зареєстрованих користувачів включає в себе ряд важливих можливостей. Вона дозволяє користувачам створювати облікові записи та авторизуватися на сайті, що забезпечує доступ до персоналізованого досвіду.

Кожен користувач має можливість реєстрації, введенням особистих даних, таких як електронна пошта та пароль. Ці дані зберігаються в базі даних і захищаються від несанкціонованого доступу[11].

На рисунку 3.3.10 зображено приклад зберігання даних користувача в базі даних.

```
_id: ObjectId('666031bba6e1e23ba763038d')
email: "nazar@gmail.com"
password: "$2b$04$emTYcCPZ7aMFBd4jI3ye/eVRhR8fVILbL1vhoXEz578M2/YIjifAS"
isActive: false
activationLink: "5932e67a-4926-44b7-86c2-8ac42d631173"
```

Рисунок 3.3.10 – Приклад зберігання даних користувача

Розроблений каталог веб-сайту, складається з ряду компонентів, що спільно працюють для відображення металопластикових конструкцій і керування замовленнями користувачів.

App - це головний компонент, який об'єднує всі інші компоненти разом. Він має стан, що включає кошик замовлень (orders), поточні товари (currentItems), повний об'єкт товару (fullItem), який відображається повноекранно (ShowFullItem), та список доступних товарів (items). Він також містить методи для додавання товарів в кошик (addToOrder), видалення товарів з кошику (deleteOrder), вибору категорії товарів (chooseCategory) і відображення повної інформації про товар (onShowItem). Структуру каталогу відображено на рис 3.3.11

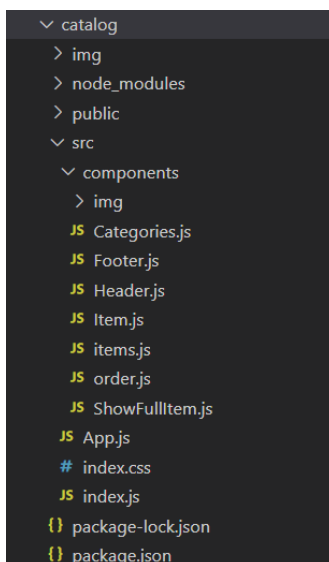


Рисунок 3.3.11 – Структура каталогу

Каталог спроектовано для зручного пошуку і вибору віконних конструкцій залежно від потреб користувача. Це зображено на рис. 3.3.12

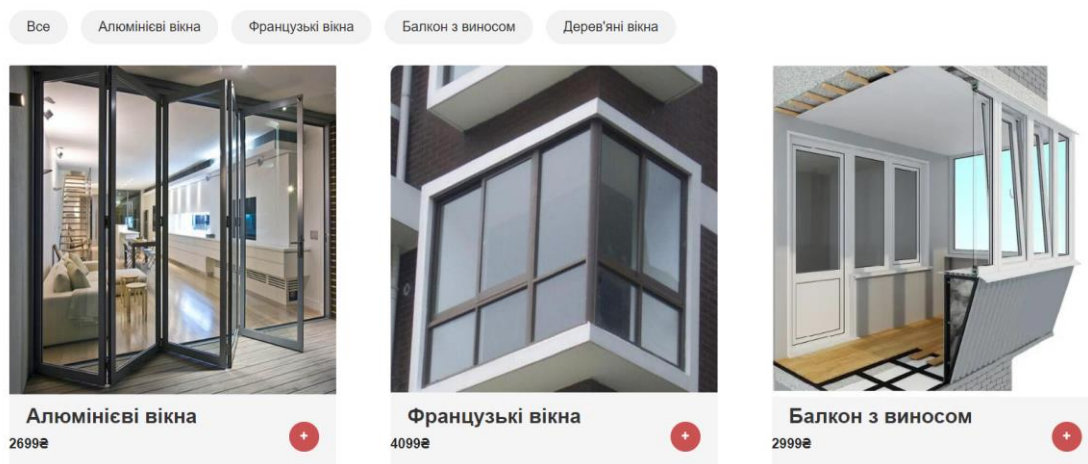


Рисунок 3.3.12 – Каталог веб-сайту

Калькулятор вартості металопластикових конструкцій на сайті має широкі можливості для користувачів. Інтерфейс дозволяє вибрати серед різних типів вікон, включаючи одноствулкові, двостулкові, трьохствулкові та інші варіанти. Користувач може вказати точні розміри вікна або дверей за допомогою зручних полів вводу та повзунків.

Окрім вибору розмірів, калькулятор також пропонує користувачам обрати профіль вікна або дверей, включаючи економ, стандарт та преміум варіанти. Кожен профіль має свої характеристики та впливає на кінцеву вартість. Після вибору профілю користувачі можуть додати до свого замовлення москітні сітки, обираючи між рамними та рулонними варіантами.

Також користувачі можуть додати підвіконня, вибравши з різних доступних типів, таких як Brusbox або Moeller, з різною шириною. Відкоси можуть бути як внутрішні, так і зовнішні, і їх вартість також враховується при розрахунку загальної вартості. Додатково, є можливість замовити монтаж конструкцій, що також відображається у кінцевій ціні.

Структура наведеного калькулятора зображена на рис 3.3.13

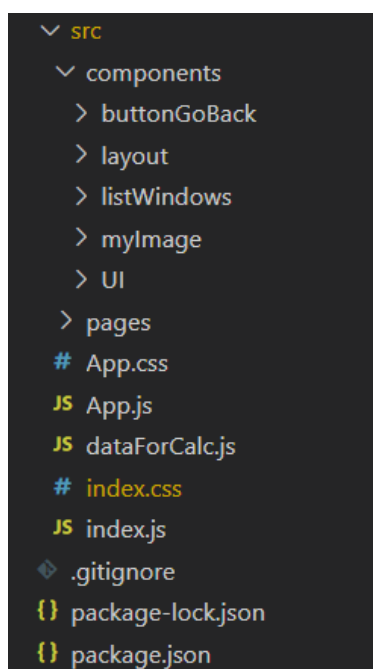


Рисунок 3.3.13 – Структура калькулятора

Інтерфейс калькулятора зображено на рисунку 3.3.14

The image shows a four-step window configuration interface:

- 1 ОБОРІТЬ ТИП ВІКНА**: A row of icons representing different window styles (single, double, triple, etc.).
- 2 ВКАЖІТЬ РОЗМІР**: Dimensions are set to 1912 mm by 1150 mm. A preview shows a window with a city view.
- 3 ВИБЕРІТЬ КОМПЛЕКТУЮЧІ**:
 - ПРОФІЛЬ: ЕКОНОМ (dropdown)
 - МОСКІТНА СІТКА: ☐ (checkbox)
 - ВІДКОСИ: ☐ Зовнішній, ☐ Внутрішній (checkboxes)
 - ПІДВІКОННИК: Ні (dropdown)
 - МОНТАЖ: ☐ (checkbox)
 - Рамна: ☐ (radio button)
 - Рулонна: ☐ (radio button)
- 4 ЦІНА ВИРОБУ**: The final price is displayed as **4397** грн.

Рисунок 3.3.14 – Інтерфейс калькулятора

Весь процес вибору та налаштування компонентів відображається в режимі реального часу, дозволяючи користувачам бачити, як змінюється ціна виробу залежно від їхніх виборів. Кінцева вартість відображається у зручному форматі, що дозволяє користувачам швидко оцінити, скільки коштуватиме обраний ними виріб.

3.4 Тестування програмного забезпечення

Тестування є важливою складовою процесу розробки програмного забезпечення, що забезпечує високу якість, надійність і відповідність системи вимогам користувачів. У роботі буде застосовано кілька видів тестування: функціональне тестування, баг-репорти та автоматичне тестування [9].

Функціональне тестування передбачає перевірку коректності роботи всіх функцій калькулятора вартості металопластикових конструкцій. Ми протестували основні функції системи, включаючи обчислення вартості конструкцій,

реєстрацію та авторизацію користувача, категорії каталогу і відправка повідомлення через контактну форму. Особлива увага приділялась валідації даних, що вводяться користувачем, і відповідності результатів обчислень очікуваним.

Тест-кейс 1: Обчислення вартості стандартного вікна

Назва тест-кейсу: Перевірка обчислення вартості стандартного вікна

Передумови: Користувач знаходиться на сторінці калькулятора вартості.

Кроки:

- 1) Обрати тип вікна "Двостулкове вікно".
- 2) Встановити розміри вікна: ширина - 1500 мм, висота - 1200 мм.
- 3) Обрати профіль "Економ".
- 4) Обрати підвіконня "Brusbox 300мм".

Очікуваний результат: Відображається правильна загальна вартість виробу, яка включає обрану конфігурацію.

Тест-кейс 2: Реєстрація нового користувача

Назва тест-кейсу: Перевірка реєстрації нового користувача

Передумови: Користувач знаходиться на сторінці реєстрації.

Кроки:

- 1) Заповнити поля "Ім'я", "Електронна пошта", "Пароль", "Підтвердження пароля".
- 2) Натиснути кнопку "Зареєструватися".

Очікуваний результат: Користувач успішно зареєстрований та перенаправлений на головну сторінку.

Тест-кейс 3: Авторизація користувача

Назва тест-кейсу: Перевірка авторизації користувача

Передумови: Користувач вже зареєстрований і знаходиться на сторінці входу.

Кроки:

- 1) Заповнити поля "Електронна пошта" та "Пароль".
- 2) Натиснути кнопку "Увійти".

Очікуваний результат: Користувач успішно авторизується та перенаправлений на головну сторінку.

Тест-кейс 4: Перегляд каталогу

Назва тест-кейсу: Перевірка перегляду каталогу продукції

Передумови: Користувач знаходиться на головній сторінці.

Кроки:

1) Натиснути на вкладку "Каталог".

2) Обрати категорію продукції.

Очікуваний результат: Відображається список продукції обраної категорії з коректними зображеннями та описами.

Тест-кейс 5: Відправка повідомлення через контактну форму

Назва тест-кейсу: Перевірка відправки повідомлення через контактну форму

Передумови: Користувач знаходиться на головній сторінці.

Кроки:

1) Заповнити поля "Ім'я" та "Телефон".

2) Натиснути кнопку "Відправити".

Очікуваний результат: Повідомлення успішно відправлене, відображається підтвердження про успішну відправку.

Ці тест-кейси допоможуть забезпечити правильне функціонування основних функцій сайту.

Баг-репорти є важливим інструментом для документування виявлених помилок під час тестування [10]. Було створено детальну систему баг-репортів, де кожна виявлена помилка реєструється з описом проблеми, кроками для її відтворення, очікуваним та фактичним результатами, а також пріоритетом вирішення. Це дозволило швидко і ефективно ідентифікувати та усувати проблеми, забезпечуючи високий рівень якості програмного продукту. В таблиці 3.4.1 наведені баг-репорти веб сайту.

Таблиця 3.4.1 – Баг-репорти веб-сайту

ID	Опис	Кроки для відтворення	Очікуваний результат	Фактичний результат	Пріоритет
1	Відсутнє повідомлення про успішне відправлення контактної форми	1.Перейти на сторінку контактів 2.Заповнити всі поля 3.Натиснути "Відправити"	Повідомлення про успішну відправку	Відсутнє повідомлення про успішну відправку	Низький
2	Некоректна робота модальних вікон на головній сторінці	1. Перейти на головну сторінку 2. Відкрити будь-яке модальне вікно 3. Виконати дію, яка повинна закрити вікно	Модальне вікно закривається	Модальне вікно не закривається	Середній

В роботі ми успішно вирішили обидва баг-репорти, які були виявлені під час тестування сайту.

Щодо багу з ідентифікатором 1, де відсутнє повідомлення про успішне відправлення контактної форми, було додано функціонал для відображення відповідного повідомлення. Після натискання кнопки "Відправити" на сторінці контактів, користувач тепер бачить спливаюче повідомлення, яке підтверджує успішне відправлення форми. Це повідомлення реалізоване за допомогою JavaScript, що гарантує миттєвий зворотній зв'язок для користувача без необхідності перезавантаження сторінки.

Щодо багу з ідентифікатором 2, де некоректно працювали модальні вікна на головній сторінці, ми переглянули та виправили JavaScript код, що відповідає за відкриття та закриття модальних вікон. Було виявлено, що проблема полягала в неправильній прив'язці обробників подій до елементів модальних вікон. Після виправлення цих помилок, модальні вікна тепер коректно закриваються після виконання відповідних дій, таких як натискання кнопки закриття або кліком поза межами вікна. Крім того, було проведене додаткове тестування для забезпечення стабільної роботи модальних вікон на різних сторінках та у різних сценаріях використання.

В процесі розробки вебсайту було використано автоматизоване тестування для забезпечення високої якості продукту та мінімізації людських помилок під час перевірки функціональності сайту. Воно включало написання та виконання тестових сценаріїв, які охоплювали різні аспекти роботи сайту, від перевірки коректного відображення елементів інтерфейсу до функціональності форм та модальних вікон.

Для автоматизованого тестування ми використовували інструмент Selenium. Він дозволив нам створювати скрипти для інтеграційного тестування, які імітують дії користувача, такі як заповнення форм, натискання кнопок та навігація по сторінках [11]. Це дало змогу автоматизувати перевірку ключових функцій сайту, таких як контактна форма та модальні вікна. Зокрема, ми створили тести, які перевіряли, чи з'являється повідомлення про успішне відправлення форми після її заповнення і відправки, а також чи коректно відкриваються та закриваються модальні вікна. Це зображено на рис 3.4.1

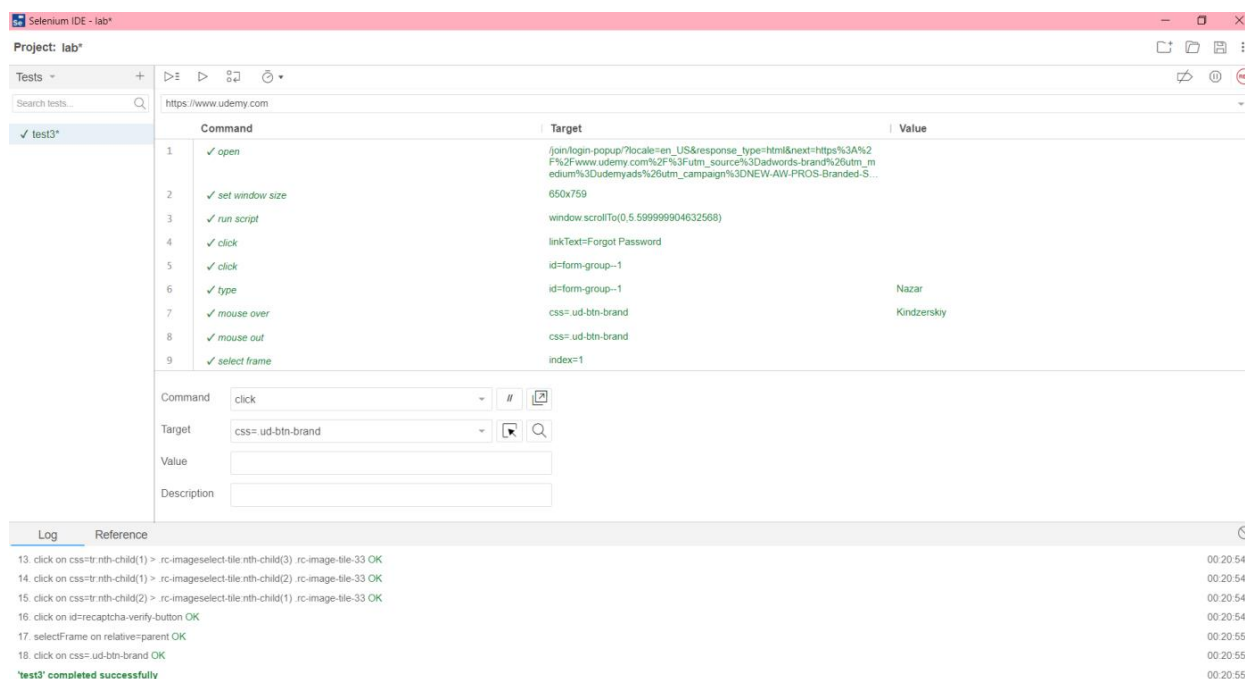


Рисунок 3.4.1 – Автоматизоване тестування

Результат використання автоматизованого тестування в проєкті був дуже позитивним і значущим. Завдяки автоматизованим тестам ми змогли забезпечити високу якість і стабільність нашого вебсайту, значно скоротивши час і ресурси, необхідні для ручного тестування.

Під час проведення фінального етапу тестування жодних критичних помилок не було виявлено. Це свідчить про високу якість коду і стабільність роботи всіх основних функцій вебсайту. Автоматизоване тестування забезпечило впевненість у тому, що продукт готовий до використання кінцевими користувачами без ризику виникнення несподіваних проблем або збоїв у роботі.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Надзвичайні ситуації екологічного характеру.

Тема "Надзвичайні ситуації екологічного характеру" присвячена аналізу та оцінці небезпек, що виникають внаслідок різних екологічних інцидентів. Цей розділ охоплює як природні, так і антропогенні чинники, що призводять до порушення екологічної рівноваги та створюють загрози для життя та здоров'я людей, тварин і рослин. Особлива увага приділяється причинам виникнення таких ситуацій, їхньому впливу на навколишнє середовище, а також заходам щодо попередження, реагування та ліквідації наслідків екологічних катастроф. Важливість даного розділу обумовлена необхідністю глибокого розуміння екологічних ризиків і розробки ефективних стратегій управління ними для забезпечення безпеки та стійкого розвитку суспільства.

Надзвичайна екологічна ситуація – це надзвичайна ситуація, при якій на окремій місцевості сталися негативні зміни в навколишньому природному середовищі, що потребують застосування надзвичайних заходів з боку держави.

До негативних змін у навколишньому природному середовищі відносять втрату, виснаження чи знищення окремих природних комплексів та ресурсів внаслідок надмірного забруднення навколишнього природного середовища, руйнівного впливу стихійних сил природи та інших факторів, які обмежують або виключають можливість життєдіяльності людини та провадження господарської діяльності в цих умовах. На сьогодні існує багато визначень суті поняття надзвичайні екологічні ситуації, до яких відноситься поняття екологічна криза та екологічна катастрофа.

Розмежування надзвичайних ситуацій, викликаних природними та техногенними факторами не завжди можливо: найчастіше характер прояву повеней, селевих потоків, снігових лавин, зсувів і багатьох інших стихійних явищ залежить як від природних процесів, так і від ступеня техногенного навантаження.

Навіть землетруси – катастрофічні події, пов'язані з процесами в земній корі і мантії, в останні десятиліття стали проявлятися в районах техногенної діяльності, які не є сейсмічно активними. Дані процеси багато фахівців пов'язують з антропогенною стимуляцією.

З іншого боку, масштаби забруднення водойм, ґрунту і повітря, опустелювання, деградації ґрунтів залежать не тільки від дій людини, але і від типу ландшафту і клімату, структури ґрунту, біотичного різноманіття. У більшості випадків антропогенні і природні фактори діють спільно, посилюючи або послаблюючи початкову дію.

Виділяють близько 70 видів небезпечних природних процесів і явищ, які можуть призвести до виникнення надзвичайної екологічної ситуації. Незважаючи на величезний технічний прогрес у розвитку суспільства, людство залишається у великій залежності від природних процесів і явищ, які щорічно приносять загибель багатьом тисячам людей, величезний матеріальний збиток. Більше того, є тенденція зростання числа жертв і економічного збитку від стихійних природних явищ, обсяги збитків від природних катастроф у світі збільшуються щорічно приблизно на 6 % [12].

Правовий режим зони надзвичайної екологічної ситуації – це особливий правовий режим, який може тимчасово запроваджуватися в окремих місцевостях у разі виникнення надзвичайних екологічних ситуацій і спрямовується для попередження людських і матеріальних втрат, відвернення загрози життю і здоров'ю громадян, а також усунення негативних наслідків надзвичайної екологічної ситуації.

У випадку запровадження в межах зони надзвичайної екологічної ситуації правового режиму надзвичайного стану відповідно до Закону України "Про правовий режим надзвичайного стану" можуть запроваджуватися, зокрема, такі заходи [13]:

- 1) встановлення особливого режиму в'їзду і виїзду;
- 2) обмеження руху транспортних засобів та проведення їхнього огляду з метою необхідної їхньої обробки, тимчасової затримки в разі виявлення

можливого небезпечного їх впливу на життя і здоров'я людей або загрози погіршення екологічної ситуації;

3) посилення охорони громадського порядку та об'єктів, що забезпечують життєдіяльність населення;

4) встановлення обмежень чи заборони на проведення масових заходів, у тому числі у разі загострення епідеміологічної обстановки крім заходів, заборона та проведення яких встановлюється судом;

5) відселення людей з місць, небезпечних для їх проживання, з обов'язковим наданням їм жилих приміщень для постійного або тимчасового проживання;

6) встановлення тимчасової заборони на будівництво нових і розширення діючих підприємств та інших об'єктів, діяльність яких не пов'язана з ліквідацією надзвичайної екологічної ситуації або із забезпеченням життєдіяльності населення;

7) встановлення карантину та здійснення інших обов'язкових санітарно-протиепідемічних заходів;

8) запровадження особливого порядку розподілення продуктів харчування і предметів першої необхідності серед населення;

9) проведення мобілізації ресурсів підприємств, установ та організацій незалежно від форм власності, зміна режиму їх роботи та профілю виробничої діяльності в межах, необхідних для проведення аварійнорятувальних та відновлювальних робіт;

10) проведення необхідних робіт по наданню допомоги тваринам у разі їх захворювання, загрози їх загибелі;

11) примусове відчуження об'єктів права приватної власності з обов'язковим наступним повним відшкодуванням їх вартості;

12) виконання комплексу робіт щодо нормалізації екологічного стану на території зони надзвичайної екологічної ситуації.

4.2 Суть та зміст управління охороною праці

Управління охороною праці – це підготовка, прийняття та реалізація рішень щодо здійснення організаційних, технічних, санітарно-гігієнічних та лікувально-профілактичних заходів, які спрямовані на збереження здоров'я та працездатності людини під час трудової діяльності. В сучасних умовах господарювання діяльність у галузі охорони праці передбачає взаємну відповідальність держави, власника підприємства (роботодавця) та працівника за створення і забезпечення здорових і безпечних умов праці, отож обґрунтованими є три рівні управління охороною праці: державне управління, управління власником підприємства та управління з боку працівників [14].

Управління охороною праці на підприємстві здійснюють роботодавець, служба охорони праці та керівники структурних підрозділів.

Роботодавець зобов'язаний створити на кожному робочому місці умови праці, які відповідають вимогам нормативноправових документів з питань охорони праці, та несе особисту відповідальність за їхню невідповідність цим вимогам.

З цією метою роботодавець:

- створює відповідні служби та призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, визначає їхні обов'язки, права та відповідальність за виконання покладених на них обов'язків, а також контролює їхнє виконання;
- розробляє за участю сторін колективного договору та реалізує комплексні заходи для досягнення встановлених нормативів та підвищення наявного рівня безпеки праці, забезпечує виконання профілактичних заходів, впроваджує досягнення науково-технічного прогресу та позитивний досвід з охорони праці;
- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці та за їхніми підсумками розробляє заходи щодо усунення небезпечних

і шкідливих виробничих чинників, забезпечує усунення причин, що спричиняють нещасні випадки;

- розробляє і затверджує положення, інструкції та інші акти з охорони праці, що діють у межах підприємства;
- здійснює контроль за дотриманням працівниками правил експлуатації обладнання, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;
- організовує навчання та перевірку знань з питань охорони праці та пропаганду безпечних методів праці.

Службу охорони праці створюють на підприємствах, на яких працює понад 50 працівників. Якщо на підприємстві працює від 20-ти до 50-ти працівників, то функції служби охорони праці може виконувати за сумісництвом працівник підприємства, який має відповідну підготовку. На підприємствах, де працює менше 20-ти працівників, для виконання функцій служби охорони праці роботодавець може залучати на договірних засадах сторонніх спеціалістів з охорони праці. Служба охорони праці підпорядкована безпосередньо роботодавцю.

Спеціалісти служби охорони праці у разі виявлення порушень мають право:

- видавати керівникам структурних підрозділів обов'язкові до виконання приписи щодо усунення виявлених недоліків;
- вимагати відсторонення від роботи осіб, які не пройшли належного медичного огляду, навчання і перевірки знань з охорони праці або не виконують вимог нормативно-правових актів з охорони праці;
- зупиняти роботу технологічного обладнання та інших засобів виробництва у разі виявлення порушень, які створюють загрозу життю або здоров'ю працівників;
- надсилати роботодавцю подання про притягнення до відповідальності працівників, які порушують вимоги щодо охорони праці.

Керівники структурних підрозділів повинні забезпечити безпечні умови праці на кожному робочому місці. Вони проводять постійний контроль за станом

засобів виробництва, засобів колективного та індивідуального захисту, станом виробничого середовища щодо його відповідності нормативним актам з питань охорони праці та вживають заходів щодо усунення виявлених недоліків. Вони також проводять навчання та перевірку знань працівників з питань охорони праці безпосередньо на робочих місцях у вигляді інструктажів та здійснюють контроль за дотриманням працівниками правил експлуатації обладнання та вимог інструкцій з безпеки праці.

ВИСНОВКИ

Розробка системи розрахунку вартості метало пластикових конструкторів стала важливим кроком у створенні сучасного та зручного інструменту для користувачів, що сприяє підвищенню ефективності та якості обслуговування клієнтів. У процесі роботи були досягнуті кілька значущих результатів, що підтверджують успішність реалізації проекту.

Калькулятор вартості метало пластикових конструкторів був розроблений з метою забезпечити користувачів зручним та швидким інструментом для розрахунку вартості необхідних виробів. Завдяки інтерактивному інтерфейсу користувачі можуть легко вибирати форму конструктора, вказувати розміри, вибирати тип скління та профіль. Це дозволяє отримати точний розрахунок вартості безпосередньо на сайті, що значно спрощує процес планування бюджету і прийняття рішень.

Каталог продукції забезпечив структуровану та зрозумілу презентацію товарів, що дозволяє користувачам швидко знаходити потрібні вироби та отримувати детальну інформацію про них. Каталог включає фільтрацію за категоріями, що дозволяє користувачам зосереджуватися на тих типах конструкторів, які відповідають їхнім потребам. Зручна навігація та інтуїтивний інтерфейс сприяють поліпшенню користувацького досвіду та підвищують рівень задоволеності клієнтів.

Головна сторінка вебсайту стала обличчям компанії, представляючи основні переваги та особливості продукції. Вона містить ключову інформацію про акції, гарантії, контакти та інші важливі аспекти діяльності компанії. Динамічні елементи, такі як таймери акцій та інтерактивні форми, сприяють залученню уваги користувачів та мотивують їх до взаємодії з сайтом.

Сайт також включає можливості для зворотного зв'язку, що дозволяє клієнтам легко зв'язуватися з компанією для отримання консультацій або уточнення деталей замовлення. Інтерактивні форми та контактна інформація на

головній сторінці сприяють оперативному обслуговуванню та підвищенню рівня довіри клієнтів до компанії. Наявність чітких і доступних форм для зворотного зв'язку забезпечує своєчасне реагування на запити та потреби користувачів, що значно покращує якість обслуговування.

Застосування сучасних технологій автоматизованого тестування забезпечило високу якість реалізованих рішень. Під час фінального етапу тестування критичних помилок не було виявлено, що свідчить про стабільність і надійність розробленого програмного забезпечення.

У підсумку, реалізація калькулятора вартості металопластикових конструкцій, каталогу продукції та головної сторінки вебсайту дозволила створити зручний, функціональний та надійний інструмент для користувачів. Це сприяє підвищенню ефективності роботи компанії, покращенню обслуговування клієнтів та зміцненню конкурентних позицій на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білас О. Є. Специфікація вимог до ПЗ [Електронний ресурс] // Національний університет "Львівська політехніка". – 2015. – Режим доступу до ресурсу: <https://comp.ucoz.net/TSP/KursRob/LPUAnalizVymog15.pdf>
2. Виробник металопластикових конструкцій Veka [Електронний ресурс] – Режим доступу до ресурсу: <https://veka.ua/ua/>
3. Зосим М. Варіанти використання та сценарії [Електронний ресурс] / М. Зосим. – 2023. – Режим доступу до ресурсу: <https://www.maxzosim.com/use-cases-and-scenarios/>
4. Опис взаємодії за допомогою діаграм послідовності [Електронний ресурс] // Таврійський державний агротехнологічний університет імені Дмитра Моторного. – 2016. – Режим доступу до ресурсу: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-9-diahrama-poslidovnosti.pdf>
5. Афонін А. О. Особливості використання документо-орієнтованих баз даних на прикладі MongoDB [Електронний ресурс] / А. О. Афонін, Д. В. Зважій. – 2015. – Режим доступу до ресурсу: <https://ekmair.ukma.edu.ua/server/api/core/bitstreams/ca708e7-e7bc-4e5e-97b1-f7dd56b2ac63/content>
6. Алгоритми JavaScript та структури даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/ukrainian/learn/javascript-algorithms-and-data-structures/>
7. Посібник по React: об'ємний путівник з вивчення React.js [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://devzone.org.ua/post/posibnyk-po-react-obyemnyy-putivnyk-z-vyvchennia-reactjs-u-2018>
8. Аутентифікація за допомогою JWT-токенів [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://krypton.com.ua/rozdil-13-autentyfikacziya-ta-avtoryzacziya/autentyfikacziya-za-dopomogoyu-jwt-tokeniv/>

9. Етапи тестування ПЗ [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/software-testing-stages/>
10. Основні атрибути баг-репорта [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/course-materials/attributes-bug-report/>
11. Автоматизоване тестування програмного забезпечення [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/avtomatizovane-testuvannya/>
12. Андронов В. А. Екологія надзвичайних ситуацій / В. А. Андронов. – Харків: Національний університет цивільного захисту України, 2021. – 147 с.
13. Закон України про зону надзвичайної екологічної ситуації [Електронний ресурс] // Верховна Рада України. – 2020. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/1908-14#Text>
14. Кіт Л. Я. Управління охороною праці [Електронний ресурс] / Л. Я. Кіт // Львівський Національний Університет імені Івана Франка. – 2018. – Режим доступу до ресурсу: https://lnu.edu.ua/life-safety/wp-content/uploads/2018/10/OP-2018_Part-3.pdf

ДОДАТКИ

ДОДАТОК А

Лістинг коду розробленого веб-застосунку

Лістинг 1 – Файл Calc.jsx

```

import {useState, useEffect, useCallback} from 'react';
import ListWindows from '../components/listWindows/ListWindows';
import MyInput from '../components/UI/myInput/MyInput';
import MyRange from '../components/UI/myRange/MyRange';
import { profiles, windowsills } from '../dataForCalc';
import MySelect from '../components/UI/mySelect/MySelect';
import MyImage from '../components/myImage/MyImage';
import axios from 'axios'

const Calc = () => {
  const [sizeOfWindow, setSizeOfWindow] = useState({
    width: 1850,
    height: 1150
  });
  const [windowType, setWindowType] = useState('twoSection1');
  const [windowImg, setWindowImg] = useState('big_two_1.jpg');
  const [rangeHorizontal, setRangeHorizontal] = useState({
    min: 1200,
    max: 2500
  });
  const [rangeVertical, setRangeVertical] = useState({
    min: 800,
    max: 1500
  });
  const [profile, setProfile] = useState('econom');
  const [profileImg, setProfileImg] = useState('econom.jpg');
  const [windowsill, setWindowsill] = useState('Brusbox 300mm');
  const [externalslope, setExternalslope] = useState(false);
  const [internalslope, setInternalslope] = useState(false);
  const [installation, setInstallation] = useState(false);
  const [mosquito, setMosquito] = useState(false);
  const [pricesOfComponents, setPricesOfComponents] = useState({
    priceSqMeter: 2000,
    priceMosquito: 0,
    priceWindowsill: 0,
    priceExternalslope: 0,
    priceInternalslope: 0,
    priceIntallation: 0
  });
  const [priceProduct, setPriceProduct] = useState(4255);

  const calculatePrice = () => {
    const {
      width,
      height
    } = sizeOfWindow;

```

```

const {
  priceSqMeter,
  priceMosquito,
  priceWindowsill,
  priceExternalslope,
  priceInternalslope,
  priceIntallation
} = pricesOfComponents;
let price = 0;
let squareMeter = (width * height) / 1000000;

price = Math.floor(squareMeter * priceSqMeter);
price += priceMosquito;
price += Math.floor(((height * 2 + width * 2) / 1000) *
priceExternalslope);
price += Math.floor(((height * 2 + width) / 1000) *
priceInternalslope);
price += Math.floor((width / 1000) * priceWindowsill);
price += priceIntallation;
setPriceProduct(price);
};

const handleSelectWindowType = useCallback((e) => {
  const value = e.target.getAttribute('data-img');
  const windowSize = JSON.parse(e.target.getAttribute('data-
size'));
  const {
    minWidth,
    maxWidth,
    minHeight,
    maxHeight
  } = windowSize;
  setWindowSize({
    width: minWidth / 2 + maxWidth / 2,
    height: minHeight / 2 + maxHeight / 2
  });
  setRangeHorizontal({
    min: minWidth,
    max: maxWidth
  });
  setRangeVertical({
    min: minHeight,
    max: maxHeight
  });
  setWindowType(value);
  setWindowImg(`${value}.jpg`);
  calculatePrice();
}, [sizeOfWindow, calculatePrice]);

const handleInput = useCallback((e) => {
  const {
    name,
    value
  } = e.target;
  setWindowSize({
    ...sizeOfWindow,
    [name]: value
  });

```

```

    });
    calculatePrice();
  }, [calculatePrice, sizeOfWindow]);

const handleSelectProfile = useCallback((e) => {
  const {value} = e.target;
  setProfile(value);
  setProfileImg(`${value}.jpg`);
  let price = 0;
  if (value === 'econom') {
    price = 2000;
  } else if (value === 'standart') {
    price = 3000;
  } else {
    price = 4000;
  };
  setPricesOfComponents({
    ...pricesOfComponents,
    priceSqMeter: price
  });
  calculatePrice();
}, [calculatePrice, pricesOfComponents]);

const handleSelectWindowsill = useCallback((e) => {
  const {value} = e.target;
  setWindowsill(value);
  let price = 0;
  if (value === 'brusbox-300') {
    price = 500;
  } else if (value === 'brusbox-550') {
    price = 750;
  } else if (value === 'moeller-300') {
    price = 800;
  } else if (value === 'moeller-550') {
    price = 1100;
  } else {
    price = 0;
  };
  setPricesOfComponents({
    ...pricesOfComponents,
    priceWindowsill: price
  });
  calculatePrice();
}, [calculatePrice, pricesOfComponents]);

const handleCheckboxExternalslope = useCallback((e) => {
  setExternalslope(e.target.checked);
  if (e.target.checked) {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceExternalslope: 200
    });
  } else {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceExternalslope: 0
    });
  };
});

```

```

    };
    calculatePrice();
  }, [calculatePrice, pricesOfComponents]);

const handleCheckboxInternalslope = useCallback((e) => {
  setInternalslope(e.target.checked);
  if ((e.target.checked)) {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceInternalslope: 300
    });
  } else {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceInternalslope: 0
    });
  };
  calculatePrice();
}, [calculatePrice, pricesOfComponents]);

const handleCheckboxInstallation = useCallback((e) => {
  setInstalation(e.target.checked);
  if ((e.target.checked)) {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceIntallation: 500
    });
  } else {
    setPricesOfComponents({
      ...pricesOfComponents,
      priceIntallation: 0
    });
  }
  calculatePrice();
}, [calculatePrice, pricesOfComponents]);

const handleRadioMosquito = useCallback((e) => {
  const {value} = e.target;
  setMosquito(value);
  let price = 0;
  if (value === 'ramna') {
    price = 500;
  } else {
    price = 750;
  };
  setPricesOfComponents({
    ...pricesOfComponents,
    priceMosquito: price
  });
  calculatePrice();
}, [mosquito, calculatePrice, pricesOfComponents]);

useEffect(() => {
  calculatePrice();
}, [calculatePrice]);

const handleSendResults = async () => {

```

```

    try {
      const result = await
    axios.post('http://localhost:5000/api/saveCalculation', {
      sizeOfWindow,
      windowType,
      profile,
      windowsill,
      externalslope: Boolean(externalslope),
      internalslope: Boolean(internalslope),
      installation: Boolean(installation),
      mosquito: Boolean(mosquito),
      priceProduct
    });
    console.log(result.data);
  } catch (error) {
    console.error('Помилка при збереженні розрахунку:', error);
  }
};

return (
  <>
    <div className='container'>
      <div className='row'>
        <div className='col-xl-6'>
          <div className='block'>
            <span className='title'>Оберіть тип вікна</span>
            <div className='number'>1</div>
            <ListWindows action={handleSelectWindowType} />
          </div>

          <div className="block">
            <span className="title">Виберіть комплектуючі</span>
            <div className="number">3</div>
            <form className="additional">
              <label htmlFor="profile">ПРОФІЛЬ</label>
              <MySelect name='profile' value={profile}
items={profiles} action={handleSelectProfile} className='list' />
              <div className="description">
                <MyImage
                  src={profileImg}
                  alt='профіль'
                  width='100%'
                />
              </div>
            </div>
            <div className="clearfix"></div>
            <div className="one">
              <label htmlFor="mosquito-net">МОСКІТНА
CITKA</label>
            </div>
            <div className="two">
              <MyInput
                type='radio'
                name='mosquito'
                value='ramna'
                className="choice"
                action={handleRadioMosquito}

```

```

        />
        <label>Пам'яна</label>
    </div>
    <div className="three">
        <MyInput
            type='radio'
            name='mosquito'
            value='roll'
            className="choice"
            action={handleRadioMosquito}
        />
        <label>Пулонна</label>
    </div>
    <br />
    <div className="one">
        <label>ВІДКОСИ</label>
    </div>
    <div className="two">
        <MyInput
            type='checkbox'
            name='externalslope'
            value={externalslope}
            checked={externalslope}
            className="choice"
            action={handleCheckboxExternalslope}
        />
        <label>Зовнішній</label>
    </div>
    <div className="three">
        <MyInput
            type='checkbox'
            name='internalslope'
            value={internalslope}
            checked={internalslope}
            className="choice"
            action={handleCheckboxInternalslope}
        />
        <label>Внутрішній</label>
    </div>
    <br />
    <div className="one">
        <label>ПІДВІКОННИК</label>
    </div>
    <div className="three">
        <MySelect name='windowsill' value={windowsill}
items={windowsills} action={handleSelectWindowsill}
className='window-sill' choose='none' />
    </div>
    <div className="clearfix"></div>
    <label>
        <MyInput
            type='checkbox'
            name='installation'
            value={installation}
            checked={installation}
            className="choice"
            action={handleCheckboxInstallation}

```

```

        />
        МОHTAЖ
    </label>
</form>
</div>
</div>

<div className="col-xl-6">
    <div className="block">
        <span className="title">Вкажіть розмір</span>
        <div className="number">2</div>
        <div className="sizing">
            <div className="w-h">
                <MyInput
                    key='width-window'
                    name='width'
                    value={sizeOfWindow.width}
                    action={handleInput}
                    className='width-window'
                />
                <span className="text">мм &#215;</span>
                <MyInput
                    key='height-window'
                    name='height'
                    value={sizeOfWindow.height}
                    action={handleInput}
                    className='height-window'
                />
                <span className="text">мм</span>
            </div>
        </div>
    </div>

    <div className="view-window">
        <MyImage
            src={windowImg}
            alt='тип вікна'
        />
        <div className="verticalSlider"
id="verticalSlider">
            <MyRange
                name='height'
                type="range"
                min={rangeVertical.min}
                max={rangeVertical.max}
                className="vertical"
                id="height-size"
                action={handleInput}
                value={sizeOfWindow.height}
            />
        </div>
    </div>

    <div className="horizontalSlider">
        <MyRange
            name='width'
            type="range"
            min={rangeHorizontal.min}

```



```

    render() {
      return (
        <div className="wrapper">
          <Header orders={this.state.orders} onDelete={this.deleteOrder} />
          <Categories chooseCategory={this.chooseCategory} />
          <Items onShowItem={this.onShowItem} items={this.state.currentItems}
onAdd={this.addToOrder} />
            {this.state.ShowFullItem && <ShowFullItem onAdd={this.addToOrder}
item={this.state.fullItem} onShowItem={this.onShowItem} />}
          <Footer />
        </div>
      );
    }

    onShowItem(item) {
      this.setState({fullItem: item})
      this.setState({ShowFullItem: !this.state.ShowFullItem})
    }

    chooseCategory(category) {
      if (category === 'all') {
        this.setState({currentItems: this.state.items})
        return
      }

      this.setState({
        currentItems: this.state.items.filter(el => el.category ===
category)
      })
    }

    deleteOrder (id) {
      this.setState({orders: this.state.orders.filter(el => el.id !==
id)})
    }

    addToOrder(item) {
      let isInArray = false;
      this.state.orders.forEach(el => {
        if(el.id === item.id) {
          isInArray = true;
        }
      })

      if(!isInArray) {
        this.setState({orders: [...this.state.orders, item] })
      }
    }
  }
}

export default App;

```

Лістинг 3 – Файл LoginForm.tsx

```

import React, { FC, useContext, useState } from "react";
import { Context } from "..";
import { observer } from 'mobx-react-lite';
import backgroundImage from "./bg.png";

const LoginForm: FC = () => {
  const [email, setEmail] = useState<string>('')
  const [password, setPassword] = useState<string>('')
  const { store } = useContext(Context);

  const handleLogin = () => {
    store.login(email, password);
  };

  const handleRegistration = () => {
    store.registration(email, password);
  };

  return (
    <div style={pageStyle}>
      <div style={formContainerStyle}>
        <h1 style={headingStyle}>Вхід в аккаунт</h1>
        <input
          type="text"
          placeholder="Email"
          onChange={(e) => setEmail(e.target.value)}
          value={email}
          style={inputStyle}
        />
        <input
          type="password"
          placeholder="Пароль"
          onChange={(e) => setPassword(e.target.value)}
          value={password}
          style={inputStyle}
        />
        <button onClick={handleLogin}
style={loginButtonStyle}>
          Вхід
        </button>
        <div style={{ marginBottom: '10px' }}></div>
        <button onClick={handleRegistration}
style={registerButtonStyle}>
          Реєстрація
        </button>
        <p style={privacyNoticeStyle}>Ваші дані
конфіденційні</p>
      </div>
    </div>
  );
};

export default observer(LoginForm);

```

ДОДАТОК Б

Диск з розробленою програмою