

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Веб-орієнтована інформаційна система формування  
кошторису будівельних робіт»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології  
проектування»

**Виконавець роботи:** студентка групи ІТ.мз-91с Марченко Ольга Вікторівна

**Кваліфікаційну роботу  
захищено на засіданні ЕК  
з оцінкою**

\_\_\_\_\_

«\_\_\_» грудня 2020 р.

Науковий керівник

\_\_\_\_\_  
(підпис)

к.т.н., доц., Парфененко Ю.В.

Голова комісії

\_\_\_\_\_  
(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Суми-2020

**Сумський державний університет**  
**Центр заочної, дистанційної та вечірньої форм навчання**  
**Кафедра комп'ютерних наук**  
**Секція інформаційних технологій проектування**  
**Спеціальність 122 «Комп'ютерні науки»**  
**Освітньо-професійна програма «Інформаційні технології проектування»**

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
«\_\_\_» \_\_\_\_\_ 2020 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

Марченко Ольга Вікторівна

(прізвище, ім'я, по батькові)

**1 Тема проекту** Веб-орієнтована інформаційна система формування кошторису будівельних робіт

затверджена наказом по університету від «16» листопада 2020 р. №1773-III

**2 Термін здачі студентом закінченого проекту** « 07 » \_\_\_\_\_ грудня \_\_\_\_\_ 2020 р.

**3 Вхідні дані до проекту** фото будинків, логотип, форма кошторису будівельних робіт, форма кошторису матеріалів, таблиці цін на види робіт та матеріали, таблиця одиниць вимірювань, форма реєстрації

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області, постановка задачі, проектування веб-орієнтованої інформаційної системи формування кошторису будівельних робіт, реалізація веб-орієнтованої інформаційної системи формування кошторису будівельних робіт.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність роботи, мета та задачі проекту, аналіз аналогів, порівняльна таблиця аналогів, функціональні вимоги, засоби реалізації, моделювання проекту (контекстна діаграма, діаграма декомпозиції), діаграма варіантів використання, модель бази даних, архітектура проекту, зовнішня структура проекту, демонстрація проекту, здійснення впровадження системи в роботу, висновки.

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів випускного проекту	Термін виконання етапів проекту	Примітка
1	Ознайомлення з предметною областю	20.05.20-26.06.20	
2	Формування мети і задач	23.05.20-26.05.20	
3	Аналіз аналогів та проблем використання	27.05.20-12.06.20	
4	Виявлення вимог до проекту	10.06.20-26.06.20	
5	Планування робіт та розробка макету	23.06.20-29.06.20	
6	Вибір засобів реалізації	28.06.20-30.06.20	
7	Проектування інформаційної системи	01.09.20-30.09.20	
8	Розробка інформаційної системи	01.10.20-02.11.20	
9	Тестування та завершення роботи	03.11.20-09.11.20	
10	Оформлення пояснювальної записки	10.11.20-01.12.20	

Магістрант \_\_\_\_\_

Марченко О.В.

Керівник роботи \_\_\_\_\_

к.т.н., доц. Парфененко Ю.В.

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Веб-орієнтована інформаційна система формування кошторису будівельних робіт».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 46 найменувань, 3 додатків. Загальний обсяг роботи – 118 сторінок, у тому числі 112 сторінок основного тексту, 6 сторінок списку використаних джерел, 27 сторінок додатків. Пояснювальна записка включає в себе 79 рисунків та 5 таблиць.

Кваліфікаційну роботу магістра присвячено розробці веб-орієнтованої інформаційної системи формування кошторису будівельних робіт. В роботі проведено аналіз предметної області, а саме огляд застосування інформаційних технологій у будівництві та інформаційного забезпечення оформлення документації по будівництву. Також розглянуто програмні продукти для формування кошторису будівельних робіт. Була поставлена задача, визначені функціональні вимоги та проаналізовані й обрані засоби реалізації продукту.

У роботі виконано проектування інформаційної системи, а саме виконано структурно-функціональне моделювання процесу формування кошторису будівельних робіт, моделювання варіантів використання розробки та проектування моделі бази даних. Описаний процес реалізації інформаційної систем.

Результатом проведеної роботи є реалізована веб-орієнтована інформаційна система формування кошторису будівельних робіт.

Практичне значення роботи полягає у легкості та швидкості ведення процесу формування кошторисів, завдяки якому заощаджуються час та кошти як компанії так і клієнтів.

Ключові слова: інформаційні технології, інформаційні системи, веб-сайт, фреймворк Laravel, MVC, будівництво, кошторис робіт, кошторис матеріалів.

## ЗМІСТ

ВСТУП .....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Огляд застосування інформаційних технологій у будівництві .....	9
1.2 Аналіз інформаційного забезпечення оформлення документації по будівництву .....	13
1.3 Огляд програмних продуктів для оформлення кошторису будівельних робіт .....	17
2. ПОСТАНОВКА ЗАДАЧІ .....	22
2.1 Мета та задачі розробки .....	22
2.2 Вибір засобів реалізації веб-орієнтованої інформаційної системи формування кошторисів будівельних робіт .....	32
3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФОРМУВАННЯ КОШТОРИСУ БУДІВЕЛЬНИХ РОБІТ .....	37
3.1 Структурно-функціональне моделювання формування кошторису будівельних робіт .....	37
3.2 Моделювання варіантів використання web-орієнтованої інформаційної системи формування кошторису будівельних .....	41
3.3 Проектування моделі бази даних .....	44
4 РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФОРМУВАННЯ КОШТОРИСУ БУДІВЕЛЬНИХ РОБІТ .....	48
4.1 Розробка архітектури інформаційної системи .....	48
4.2 Встановлення та розробка компонентів інформаційної системи .....	50
4.3 Результат реалізації веб-орієнтованої інформаційної системи формування кошторису будівельних робіт .....	59
ВИСНОВКИ .....	86

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	88
Додаток А.....	94
Додаток Б .....	101

## ВСТУП

Інформація та комп'ютеризація є одними з важливих ресурсів сьогодення. При порушенні надходженні інформації відбувається виникнення проблем в роботі пристроїв, процесів і навіть великих систем господарства.

Техніка для обчислювання все ширше використовується в технологічному та промисловому процесах й у всій економіці світу. Комп'ютеризація та нові інформаційні технології (ІТ) допомагають вдосконалювати діяльність, тому що вони базуються на своєчасній та точній інформації, а також забезпечують швидкий її обмін. Стратегічними цілями інформаційних технологій є забезпечення розвитку бізнесу, його керування і якість, конкурентоспроможність та зниження вартості виконання бізнес-процесів.

Не є виключенням і будівельна галузь, для управління якої необхідно проводити розрахунки робіт, матеріалів та затрат на різних етапах роботи.

На сьогодні існує багато розрахункових програм, але більшість з них є платними та мають функціональність розраховану на великі будівельні компанії зі складною системою обрахунків. Це не полегшує, а навпаки ускладнює обчислення та складання кошторисів для малих підприємств (Limited Liability Company – LLC). Система повинна працювати на власника, а не навпаки. Час є цінним, тому у сучасному світі при великій мобільності успішні компанії намагаються за короткий час виконати максимально більше завдань, щоб досягти максимально більших результатів. Таким чином, актуальною є задача розробки гнучкої інформаційної системи формування кошторисів будівельних робіт, що буде враховувати усі вимоги замовника.

Об'єкт дослідження: використання інформаційних технологій у будівельній галузі.

Предмет дослідження: інформаційне забезпечення формування кошторису будівельних робіт.

Метою кваліфікаційної роботи магістра є розробка веб-орієнтованої інформаційної системи формування кошторису будівельних робіт. Для досягнення мети проекту ставимо наступні задачі:

- провести аналіз проблеми використання інформаційних технологій при оформленні документації по будівництву;
- провести аналіз існуючих програмних продуктів, що використовуються при формуванні кошторису будівельних робіт, визначити вимоги до проекту та виконати планування робіт;
- провести аналіз технологій розробки веб-орієнтованих інформаційних систем для підтримки ведення будівельної документації та обрати засоби реалізації;
- виконати моделювання процесу формування кошторису будівельних робіт з використанням веб-орієнтованої інформаційної системи;
- розробити веб-орієнтовану інформаційну систему формування кошторису будівельних робіт та провести її тестування.

Опираючись на поставлені задачі на завершальному етапі, як результат, буде розроблено інформаційний продукт, який дозволить швидко та практично отримати оцінюючу документацію будівельних робіт на місці складання проекту, обговорюючи всі важливі моменти з клієнтом, а також дозволить систематизувати дані кошториси, що полегшуватиме звітність перед податковою інспекцією.



## 1 АНАЛІЗ ПРЕДМЕТНОГО ОБЛАСТІ

### 1.1 Огляд застосування інформаційних технологій у будівництві

Змінюється інформаційний світ, змінюється життя, а саме система цінностей. Матеріальність відходить на другий план, віддаючи перевагу інформації, часу та культурі. Пріоритет розумової праці в інформаційному суспільстві супроводжується виробленням та споживанням знань та інтелекту. Тому значне місце займає здатність до творчого мислення, яке породило низку інформаційних технологій [1].

У сучасному світі використання ІТ відбувається в усіх сферах людської діяльності. Інтелектуальні системи та програми виконують першочергові функції в бухгалтерії, економіці, фінансах, рекламі, музиці, освіті, медицині, архітектурі і багатьох інших напрямках. На сьогоднішній день не є виключенням будівельна галузь, яка також має великий прогрес розвитку завдяки використанню ІТ. Спостерігаються значні позитивні зміни в роботі таких фахівців, як будівельника, дизайнера, архітектора, менеджера управління та для замовника. Комп'ютеризація допомагає від самого початку зародження ідеї до створення проекту та його реалізації, візуалізуючи результат, складаючи розрахунки і кошториси, зводячи конструкції та управляючи самим об'єктом [2].

У галузі будівництва використання інформаційних технологій розпочалося з виконання завдань різних розрахунків. На сьогоднішній день воно трансформувалося в систему управління складними проектами, розпочинаючи із звичайних будівель, комунікаційних систем і закінчуючи автоматизованими системами управління об'єктами державного нагляду [3].

Так основним сектор інформаційних технологій в сфері будівництва є програмне забезпечення процесу проектування та зведення будівлі з

можливістю відстеження стану будівництва на конкретних етапах, забезпечення багатоваріантності проектів (завдяки змінні та варіації вихідних даних), збільшення якості проектування, зменшення кількості витрат на матеріали, інженерних працівників, зайнятих конструюванням і проектуванням, скорочення строків проектування [4, 5].

У сучасному великому будівельному бізнесі все більш активно використовують ІТ та спеціалізоване програмне забезпечення (ПЗ). Перш за все це система автоматизованого проектування (САПР) [6]. Вони допомагають виконувати:

- архітектурне планування;
- вирішувати задачі планування проекту;
- дизайнерські питання;
- розрахувати механічні характеристики споруд;
- управляти процесом самого будівництва [6].

Найпопулярнішими програмними продуктами являються Allplan, AutoCAD, ArchiCAD, SCAD Office, nanoCAD, Revit.

ArchiCAD вважається кращим додатком, що застосовуються в будівельно-архітектурному проектуванні. Інформаційні технології в програмі ArchiCAD дозволяють побудувати віртуальну модель реальних конструкцій, завдяки використанню інструментів, що мають реальні аналоги (колони, стіни, вікна, перекриття і так далі) [7]. Будь-які зміни зроблені, наприклад на плані будівлі, автоматично відображаються на розрізах, в специфікації, експлікації і т.п. Такі підходи забезпечують значне скорочення часу проектування. Одночасно формується документація [8]. Також в програмі є і недоліки, які потрібно допрацьовувати:

- обмежені можливості для створення об'єктів зі складною нестандартною геометрією (наприклад скульптурне модулювання);
- не передбачає багатоваріантність проектування;
- достатньо висока ціна (\$ 1500) [9].

Існують інформаційні системи, які призначені для управління підприємства в будівництві. До них відносять «ІС: Управління будівельною організацією», «ІС: Підрядник будівництва. Управління фінансами». Системи допомагають у складанні календарних планів, контролі за виконанням робіт. Є можливість проводити обмін даними з кошторисними і фінансовими програмами [10].

Таким чином, значне місце займає моделювання. Сучасний замовник хоче мати не тільки якісну будівлю, але й щось довготривале, нестандартне та з мінімальними витратами. Щоб вирішити ці та інші питання використовують технології інформаційного моделювання у будівництві (BIM – Building Information Modeling) [11]. Основна технологія – це трьохвимірна модель. В залежності від задач, які будь виконуватися по ходу роботи, додаються додаткові вектори: 4D – час, 5D – вартість, 6D – експлуатація [12].

Переваги BIM є очевидними, на відміну від попередніх розробок проектування будівельного об'єкта, які створювали геометричні образи, BIM моделювання створює цифрову модель, яка включає повну інформацію про об'єкт та про процес його будівництва. Використання даної технології проектування робить кожен крок прозорим і забезпечує повний контроль та гарантує високу якість проектно-будівельних робіт [13].

Зазначимо, що все вищеописане стосується великих будівельних компаній, які займаються реалізацією зведення масштабного будівництва. Щодо малих підприємств, які ведуть діяльність спрямовану на роботи пов'язані з реставрацією будівель, ремонтами приміщень мають інші потреби в інформаційних технологіях, а саме в тих розробках, які відповідають за розрахунки кошторисів виконання робіт та за кошториси робочого матеріалу.

Інформаційні технології в будівництві дозволяють автоматичну перевірку розрахунків та створення форм для друку, тим самим полегшуючи подібну роботу, скорочуючи час на її створення та виключаючи наявність помилки [14].

Існують системи, що допомагають створювати відповідні кошториси. Найпопулярніші: МСмета, будівельні онлайн калькулятори, Будівельні Технології, Кошторис8, СМЕТА АС-4 в Україні, Construction Estimating & Job Costing Software, JobNimbus, UDA ConstructionOnline, Esticom, РМО в США та інші [15].

Неможливо не відмітити значимість застосування веб-сайтів як продукту рекламного значення для будівельного сектору. Адже саме веб-сайти набирають стрімкий ріст в інформаційній індустрії. Велика частина представників бізнесу вже не уявляють їх діяльність без даного інструменту. Інформаційні веб-сайти замінюють контактне спілкування з клієнтами, так як вони мають можливість дізнатися про компанію не тільки з інформаційного блоку, де описана діяльність організації, її досягнення та результати у вигляді фото- та відеозвітах, але й переглянути відгуки осіб, які вже скористалися послугами компанії. Якщо потенційний клієнт має додаткові питання, в нього є можливість зв'язатися по телефону, який також зазначений в контактній інформації. Поширеною опцією є реєстрація та створення свого особистого кабінету, де вже відбувається процес введення замовлення та втілення його в реальність. Все це підвищує статус компанії, збільшує коло цільової аудиторії, а також розповсюджує інформацію про діяльність та послуги.

Інформаційні технології все більше поширюються в усіх сферах життєдіяльності людини. Використовуються різноманітні форми і методи (комп'ютерна програма, веб-сайт, соціальні мережі), які вирішують серйозні, специфічні завдання. Знайшло своє місце ІТ і в будівельно-архітектурній галузі. Сучасні сервіси виступають помічниками, як для вже сформованих спеціалістів, так і для студентів, які знаходяться на етапі свого становлення. Клієнти, використовуючи інтернет, також вирішують чимало питань пов'язаних з ремонтом квартир та будинків. Арсенал ІТ постійно удосконалюється, приймаючи нові форми, для прискорення роботи, роблячи її результат кращим, щоб скоротити витрати та інше [2].

## **1.2 Аналіз інформаційного забезпечення оформлення документації по будівництву**

У попередньому підрозділі було розглянуто застосування інформаційних технологій у будівництві та виявлено їх суттєвий вплив на дану галузь. Тепер більш конкретно зупинимося на сфері малого будівельного бізнесу, який виконує менш масштабні роботи, але має широку популярність серед людей середнього класу.

Жодне будівництво не розпочинається без оцінки вартості будівельних робіт, так як потрібно дати попередню оцінку про затрати, які підуть на даний об'єкт.

Для того, щоб отримати обговорювану вартість, необхідно враховувати всі роботи і ресурси, які будуть задіяні в будівництві. Складений кошторис повинен бути виконаний згідно регламентуючих документів. Складанням кошторисів займаються інженери-кошторисники, які мають профільну освіту і вивчили як мінімум основи кошторисної справи [16].

Розглянемо процес будівництва та процес формування кошторису робіт і матеріалів через веб-сайт (рис 1.1).

На початку процесу клієнт звертається до компанії напряду чи через веб-сайт. Надається запит щодо будівельних робіт, відбувається обмін інформацією, пов'язаний з потребами і побажаннями замовника та умовами й ціновою політикою виконавця. Дані потенційного замовника виносяться в клієнтську базу.

Наступним кроком є огляд об'єкту. Представники двох сторін обговорюють на місці будівництва конкретні роботи, зміни в плануванні. Як правило, проектні роботи коригуються до і після консультації зі спеціалістами. Інформація надсилається замовнику на електронну пошту. Клієнт вносить додаткові зміни до проекту.

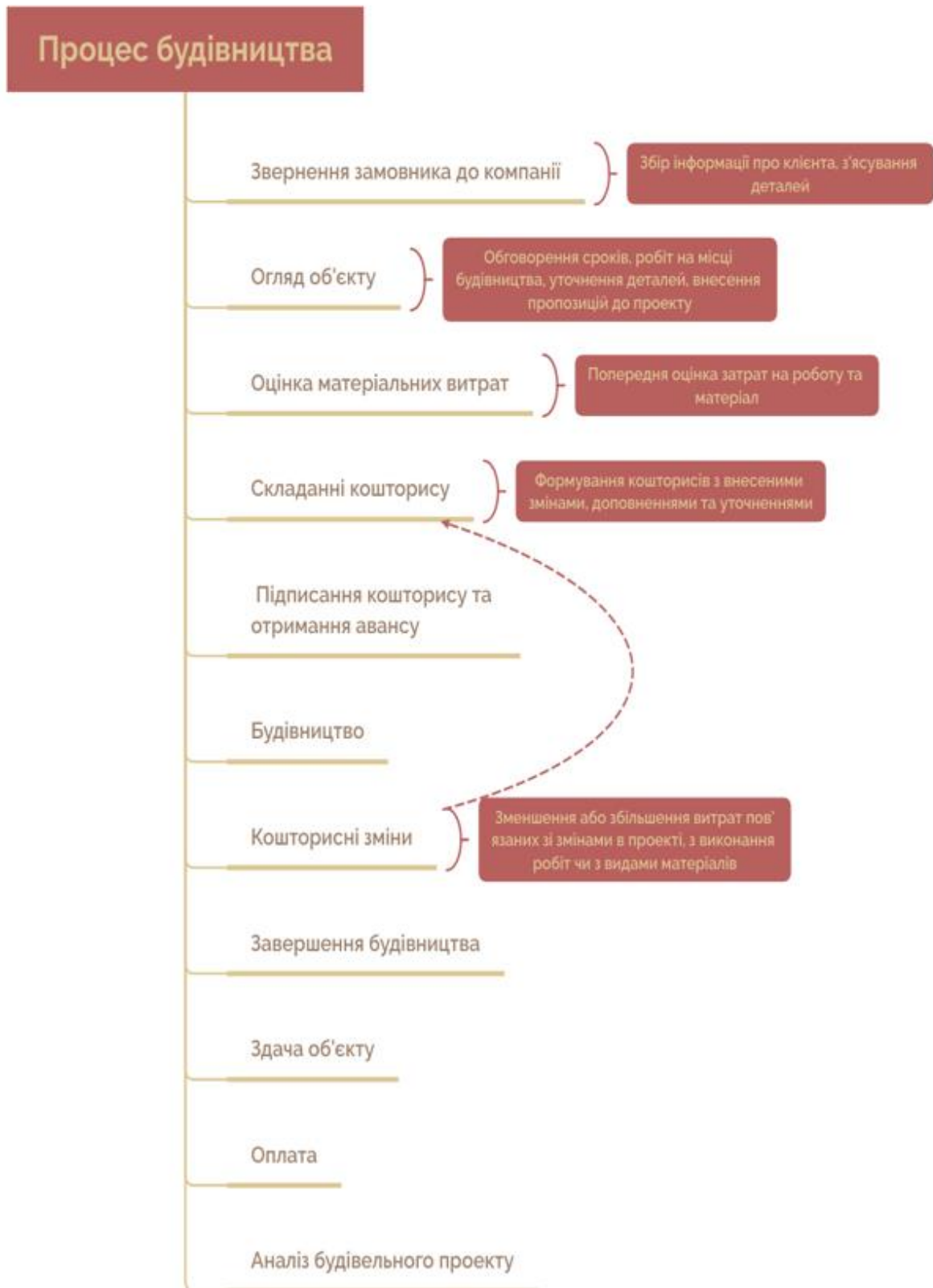


Рисунок 1.1 – Процес будівництва

Коли зміни та доповнення внесені, дві сторони підписують кошторисні проекти та вноситься перша інвестиція в вигляді авансу і розпочинається будівництво. Під час робочого процесу існує велика ймовірність того, що виникнуть додаткові коригування кошторисів, тому ведуться звітування про відхилення від початкового варіанту проекту. А також ведеться визначення авансових виплат або зміни до тривалості будівництва [17].

По завершенню будівництва об'єкт здається замовнику, перевіряється, при необхідності відбувається корекція не прийнятих робіт. Якщо проект схвалено, ведеться розрахунок між замовником та виконавцем.

Не мало важливим моментом є проведення аналізу виконаного будівельного процесу. На цьому етапі проводиться оцінка якості, технології та ходу робіт, безпеки будівництва, матеріальних збитків, економії витрат та інше.

На рис. 1.2 показана дворівнева інформаційна структура обробки даних при оформленні документації по будівництву. Під первинними та звітними документами маються на увазі попередні та погоджені кошториси робіт та матеріалів, які повинні формуватися автоматично залежно від обраних форм. Таким чином, для ефективної та оперативної обробки даних необхідно розробити інформаційну систему.

Як відомо, будь-яка система документообігу є складною, у тому числі і будівельна. Тобто, всю інформацію та всю сукупність зв'язків між елементами системи не можливо сприйняти вцілому. Тому необхідно поділити систему на простіші частини – інформаційні структури. Це зменшить складність, дозволить налагоджувати зв'язок і проявляти гнучкість, спростить кожен крок ведення будівельного процесу та управління ним (рис. 1.2.) [18, 19].



Рисунок 1.2 – Дворівнева інформаційна структура обробки даних при оформленні документації по будівництву

Первинні та звітні документи формуються на основі введених користувачем даних. Частина даних може вводиться безпосередньо на об'єкті будівництва при оформленні замовлення, тому необхідно розробити веб-орієнтовану інформаційну систему для складання кошторису будівельних робіт, щоб забезпечити віддалений доступ до даних.



### **1.3 Огляд програмних продуктів для оформлення кошторису будівельних робіт**

Паперові носії, збір інформації вручну залишаються в минулому. Складання кошторису вважається найважливішим етапом будь-яких будівельних чи ремонтних робіт. На сьогоднішній день існує велика кількість програмних систем, що дозволяють формувати оцінюючий процес будівництва. Перед кошторисником стоїть не просте завдання у виборі програмного комплексу, яким він буде користуватися в своїй повсякденній роботі, адже у кожного програмного комплексу є свої переваги і недоліки [16].

Існує кошторисна онлайн програма МСмета [20]. На сайті пропонується ознайомитися з переліком документації, яка потрібна при будівництві за бюджетні кошти. Існує можливість замовити кошторисну документацію або скласти її самостійно. Кошторис складається для України згідно ДСТУ Б Д.1.1-1:2013 "Правила визначення вартості будівництва" та розрахована для великих компаній. Цінова вартість матеріалів на сайті вказана за 2016 р.

Кошторис 3 [21]. Демо-доступ до продуктів 1С:Підприємство згідно ліцензії можливий за попереднім запитом. Після чого можна протестувати демо базу онлайн. Кошторис 3 також призначений для великих організацій. Користувачам надається велика кількість можливостей для складання власного кошторису, таких як створення особистих фрагментів і шаблонів, коригування існуючих та створення нових збірників розцінок, можливість ведення кошторисної документації по частинах, розділах і підрозділах, робота зі складом ресурсів, заміни ресурсів і т.д. Для використання програми потрібен час та проходження ознайомлювального курсу.

Але в випадку, коли будівництво ведеться в країні, де існують інші розцінки, інша вимірююча система, де існує інша законодавча база використання вище зазначених систем є неможливою. Звернемо увагу на

програми формуванням оцінки будівельних робіт, які використовуються в Сполучених Штатах Америки [22].

COCONSTRUCT – одне із рейтингових програмних забезпечень в оцінці будівництва. Воно пропонує найбільшу кількість можливостей та переваг за доступною ціною в США. У тарифному плані вказана базова місячна ставка, і всі тарифи дозволяють необмежену кількість користувачів. Програмне забезпечення включає необмежену кількість перспективних та кошторисних проектів. Якщо протягом одного місяця клієнт був не задоволений своєю послугою чи досвідом, то він отримує повне відшкодування. Вартість залежить від пакету послуг. Існує зручна комунікація з членами команди та клієнтами. Немає безкоштовної пробної версії. Обслуговування клієнтів відповідає високим рейтингам, а саме якісна консультація щодо роботи програми. Підтримку можливо отримати по телефону або надіслати електронний лист експерту при наявності проблем. Присутній доступ до живого чату. Компанія надає багато можливостей для практичного навчання: корисні вебінари та відео, які є безкоштовними [23].

STACK – це відносно недороге програмне забезпечення для оцінки будівництва. Безкоштовний план обслуговування підходить для малих підприємств. STACK забезпечує необмежену підготовку та підтримку незалежно від плану обслуговування. Окрім безкоштовного тарифного плану, є платні тарифи, за якими можна оплатити. STACK доступний у чотирьох планах обслуговування – безкоштовному та трьох платних. За платні тарифи STACK стягує щорічну плату залежно від кількості користувачів. У безкоштовний план на користувача на рік включені функції: дозволено два паралельних проекти, семиденний доступ до проекту, 10 вимірювань на проект та багатокористувацьку спільну роботу в режимі реального часу. У платному тарифі на користувача на рік включено наступні функції: необмежений доступ до проекту та вимірювання для кожного проекту, опції імпорту та експорту користувацьких баз даних, управління кодами витрат,

звіти про витрати праці та матеріальних витрат, калькулятор вартості проекту, функцію єдиного входу, спеціальні послуги з побудови елементів та навчання на місці, можливість отримати доступ до STACK з будь-якого комп'ютера, планшета чи мобільного пристрою, підключеного до Інтернету робить співпрацю простою. Оскільки STACK – це хмарний додаток, оновлення та вдосконалення отримуються автоматично, не вимагаючи жодних дій.

STACK має кілька недоліків. План безкоштовного обслуговування має багато обмежень, які ведуть до підключення платних тарифів [24].

Acumatica Cloud ERP – це хмарна та мобільна технологічна платформа, розроблена для клієнтів середнього класу з повним переглядом у реальному часі бізнесу в будь-який час, в будь-якому місці та на будь-якому пристрої з браузером. Комплект інтегрованих програм управління бізнесом Acumatica включає бухгалтерський облік та фінансовий менеджмент, розподіл, планування та контроль виробництва, часу та витрат, облік проектів, будівництво, та управління клієнтами.

Усі можливості повністю інтегровані, побудовані із використанням стандартних інструментів Microsoft і мають звіти про самообслуговування, управління документами та робочі процеси затвердження. Користувачі можуть пристосувати свою робочу область та інформаційні панелі. Клієнти можуть застосувати пакет на існуючих серверах або перейти на сучасну хмарну реалізацію [25, 26].

Розглянемо порівняльну характеристику останніх трьох будівельних ПЗ у табл. 1.1

Таблиця 1.1. Порівняння програмних продуктів

	<b>CoConstruct</b>	<b>STACK</b>	<b>Acumatica Cloud ERP</b>
<b>Безкоштовно пробний період</b>	Ні, тільки demo	Ні, тільки demo та безкоштовна версія	Ні, тільки demo

Продовження таблиці 1.1

	<b>CoConstruct</b>	<b>STACK</b>	<b>Acumatica Cloud ERP</b>
<b>Безкоштовні версії</b>	Ні	Так (велика обмеженість у використанні)	Ні
<b>Ціноутворення</b>	\$49 За перші 2 місяці, потім в залежності від тарифу від \$199 до \$299 за місяць	Тарифний план від \$999 до \$5999	В залежності від платформи і плану. Від \$15000 до \$40000
<b>Необмежена кількість користувачів</b>	Так	Ні	Ні
<b>Розгортання</b>	Cloud, web	Cloud, web	Cloud, web
<b>Інтеграція</b>	QuickBooks, Xero	Dropbox, Evernote, Google Drive, OneDrive, project management software	Dropbox Business, Microsoft
<b>Простота використання</b>	Потребує навчання	Потребує навчання	Потребує навчання
<b>Великий вибір послуг</b>	Так	Так	Так

Таким чином, будівельні ПЗ на сьогоднішній день є дуже розвинутими та гнучкими. Всі вони багато в чому схожі між собою. Більшість з них орієнтована на великі будівельні компанії, які займаються масштабним будівництвом. Звичайно, подібні проекти потребують ретельних та точних

розрахунків, які прямо впливають на прибутки компанії та її рейтинги серед конкурентів. Тому заощаджувати на подібних програмах їм не вигідно, хоча ціни на описані ПЗ є достатньо високими. Але в розрізі малих компаній ціна є ірраціональною.

Малий бізнес, особливо в цьому році, зазнає великих втрат. Звичайно, дозволити використовувати подібні будівельні програмні забезпечення з великим доступом до різноманітного функціоналу, який буде використовуватися частково, не є практичним рішенням. Навіть звернувшись до демо або безкоштовної версії, це не вирішуватиме поставлених питань малих компаній у будівництві, так як версії обмежені в використанні як по часовим проміжкам так і по функціональності [27].

Впровадження великих програмних систем, що автоматизують велике коло видів робіт будівельних підприємств є досить проблематичне та займає багато часу, дорого обходиться, а також після вимагає досить високої кваліфікації користувачів. Виходячи з цін вищеописаних продуктів доцільно використовувати програмний продукт, який може бути написано під певну організацію. Тому основним рішенням для досягнення оптимально-вигідного результату є виготовлення власної розробки, яка орієнтована на підприємство власника.

## 2. ПОСТАНОВКА ЗАДАЧІ

### 2.1 Мета та задачі розробки

Метою кваліфікаційної роботи магістра є розробка веб-орієнтованої інформаційної системи формування кошторису будівельних робіт.

Проект розробляється для невеликого будівельного підприємства, розташованого в США. Оптимальним варіантом рішення є розробка веб-сайту. Для виконавця будівельного проекту важливим являється створення попереднього кошторису на об'єкті з клієнтом, економлячи власний час та підтверджуючи свій професіоналізм.

Цільовою аудиторією веб-орієнтованої інформаційної системи є, з однієї сторони, особи, які зацікавлені в будівельно-ремонтних послугах, тобто потенційні клієнти-замовники, з іншої – виконавець робіт (будівельник, бухгалтер), спеціалісти, які формуватимуть та використовуватимуть кошторис, і адміністратор, який буде працювати з базою даних та підтримкою сайту.

Веб-сайт повинен бути доступним в мережі інтернет, легким у використанні, як для виконавця так і для замовника. Адміністратор повинен мати особливі технічні навички в експлуатації та підтримці веб-сайту, так як інформаційна система реалізується за допомогою HTML, CSS (web-сайт), PHP, Laravel (серверна частина), JS, Vue.js (клієнтська частина) та MySQL (база даних), де буде зберігатися вся інформація. Зі сторони інших користувачів достатньо мати загальні навички роботи з персональним комп'ютером і веб-браузером. Веб-сайт повинен бути написаний на англійській мові.

Веб-сайт повинен бути з авторизованим доступом та мати розмежуванням на дві частини – клієнтську та адміністратора.

Для клієнта веб-сайт повинен мати інформативно-ознайомлювальний характер. Тобто, відвідуючи сайт, потенційний замовник ознайомлюється з діяльністю компанії, з результатами виконаних робіт, відвідує сторінку з реальними відгуками про надані послуги. При зацікавленості клієнт може скористатися контактною інформацією, а саме номером телефону, електронною адресою або відвідати офіс за вказаною адресою, орієнтуючись по Google Map.

На рис. 2.1 зображена карта веб-сайту, де відображені зв'язки між його розділами та підрозділами.

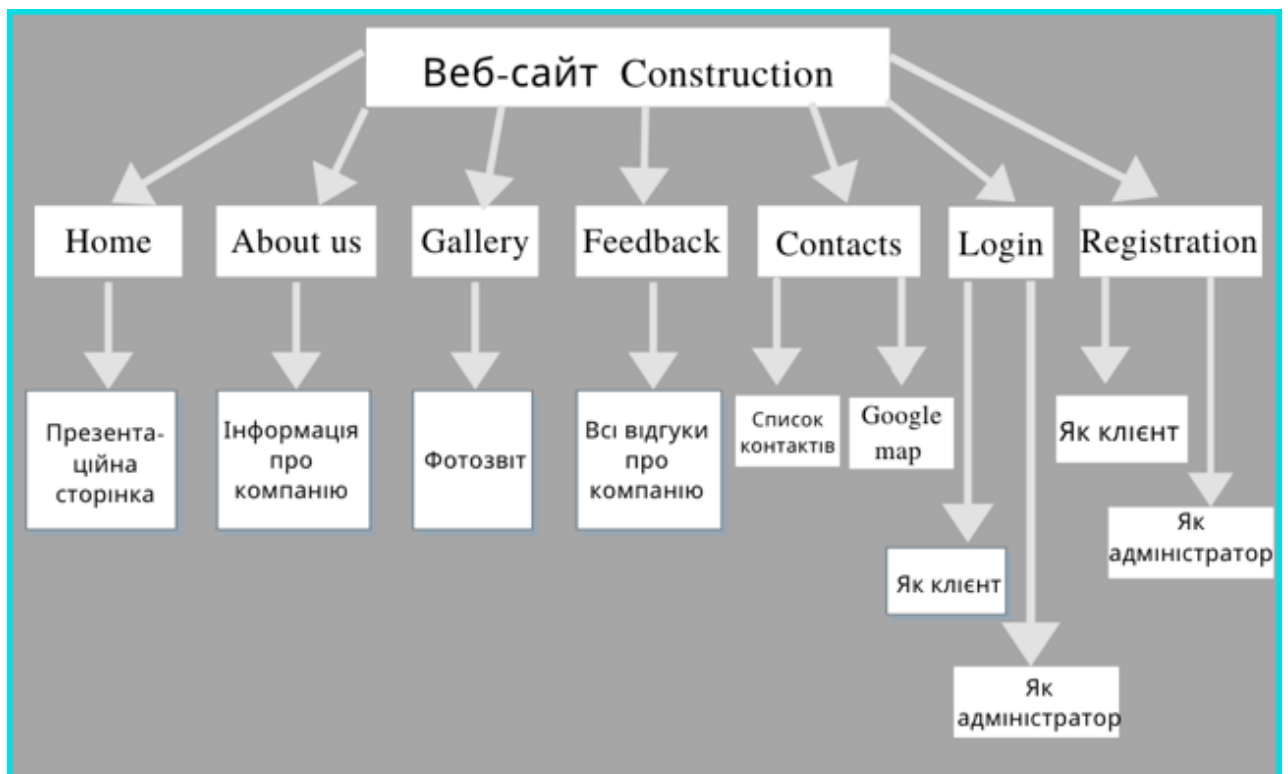


Рисунок 2.1 – Карта веб-сайту

Якщо особа хоче скористатися послугами підприємства, вона самостійно або за допомогою представника компанії реєструється на сайті, створюючи власний кабінет користувача. На рис. 2.2 зображена карта аккаунту клієнта. В ньому міститься наступне меню (Dashboard):

- My projects
- Feedback
- Price of materials

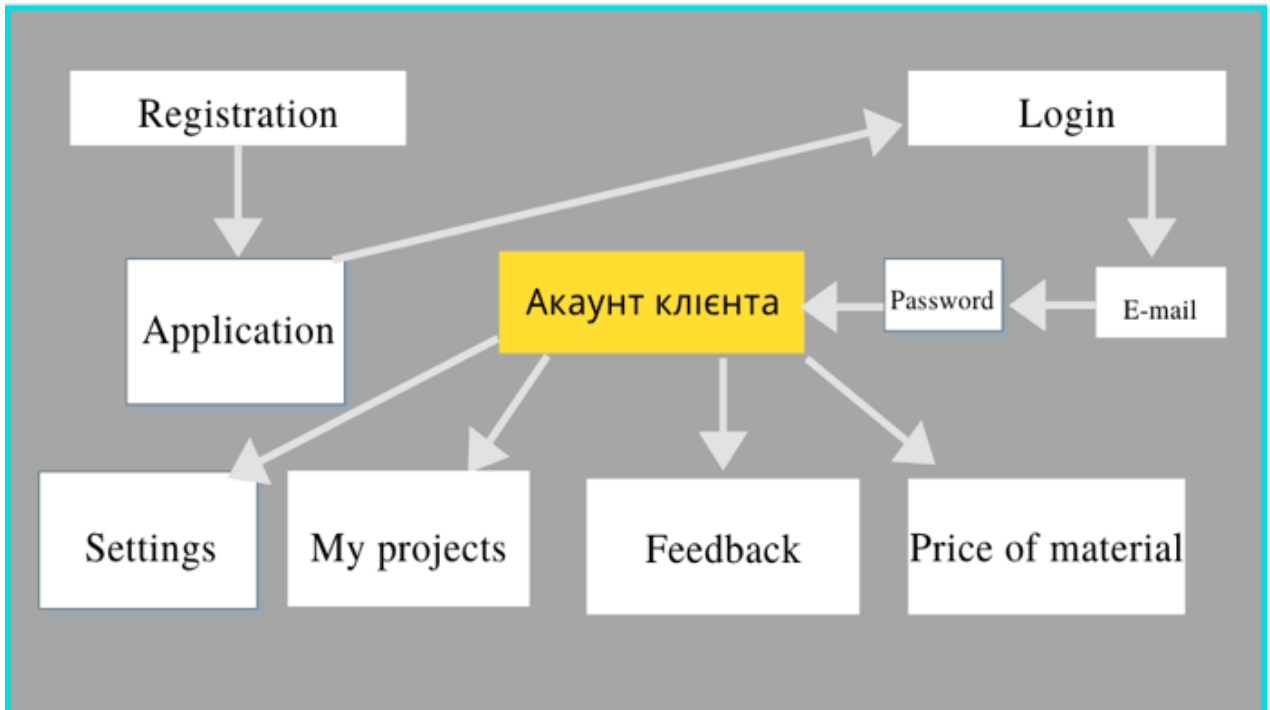


Рисунок 2.2 – Карта клієнтського акаунту

My projects – меню, в якому знаходяться всі проекти, надіслані виконавцем. Кожен проект матиме в собі кошториси на виконання робіт та на будівельні матеріали (Estimates of Work & Estimates of Materials). Новостворений кошторис матиме статус NEW (новий). Якщо клієнт його погодить він змінить статус на – in process (запустити в процес). Коли проект буде виконано, статус зміниться виконавцем на – Done (виконано), що також буде відображатися в кабінеті замовника.

Feedback – меню, в якому користувач може залишити свій відгук про виконану роботу. Описана оцінка відразу відобразиться на сайті, вона не підлягатиме корегуванню адміністратором, що говоритиме про прозорість компанії.



Price of materials – меню, в якому клієнт ознайомиться з цінами на матеріал, взяті з 2х основних будівельних магазинів Lowes та The Home Depot. На сторінці буде створено пошукову систему для швидкого знаходження потрібних одиниць, фільтр цін та фільтр списків.

Для коригування власних даних (прізвище, ім'я, по-батькові, номер телефону, електронна адреса, адреса проживання та ін.), які будуть відображатися в кошторисах, користувач зможе звернутися до налаштувань профілю (Settings). Тут же можна буде змінити пароль, який має бути зашифрований в базі даних, що забезпечує конфіденційність кабінету.

Крім аккаунту клієнта має бути створений кабінет виконавця – аккаунт адміністратора, в якому проходитиме основний процес формування кошторису будівельних робіт. На рис. 2.3 зображена карта аккаунту адміністратора.

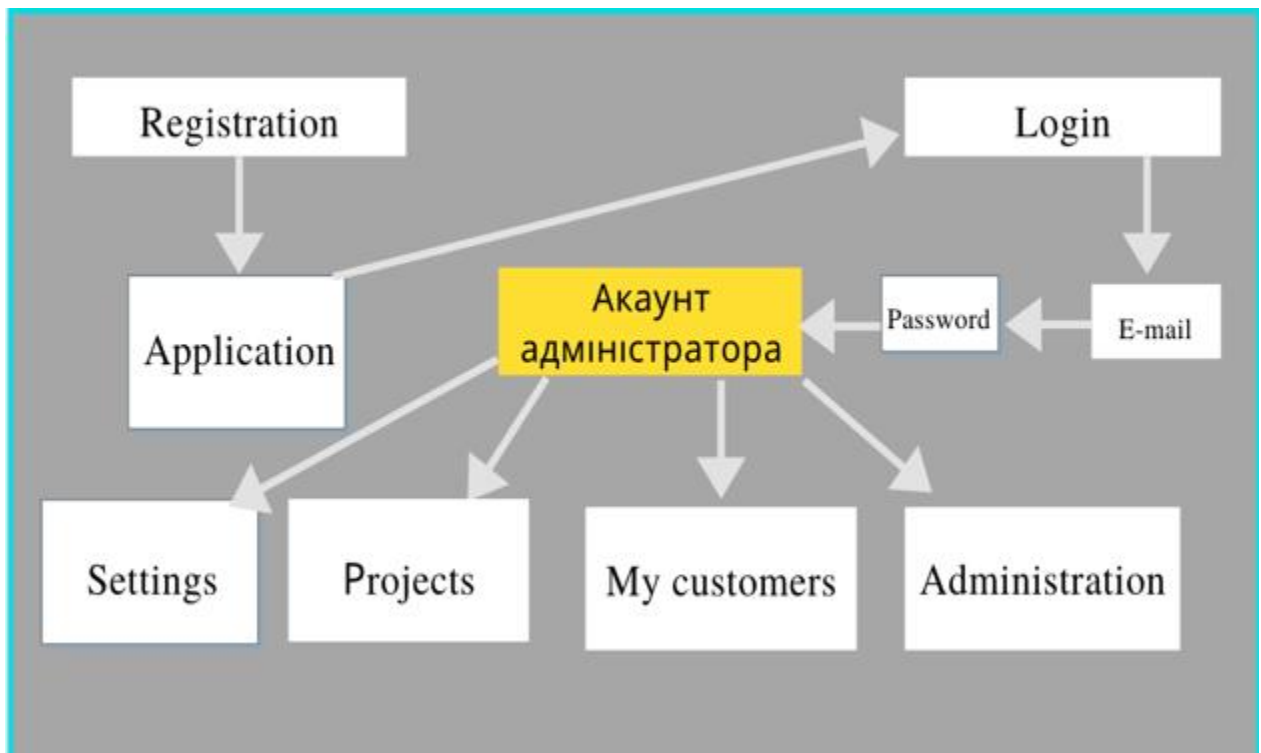


Рисунок 2.3 – Карта аккаунту адміністратора

Меню кабінета складається з:

- Projects
- My Customers
- Administration

Projects – на даній сторінці будуть створюватися нові і зберігатися раніше створені проекти в хронологічному порядку, які коригуються або видаляються. Обираючи проект, відкриватиметься нова сторінка – Project management, де розташовані 4 опції: створення кошторису робіт (Make Estimate of Work), створення кошторисів по матеріалам (Make Estimate of Materials), кошториси робіт (My Estimates of Work) та кошториси по матеріалам (My Estimates of Materials). Карта блоку Projects (Проекти) зображена на рис.2.4.

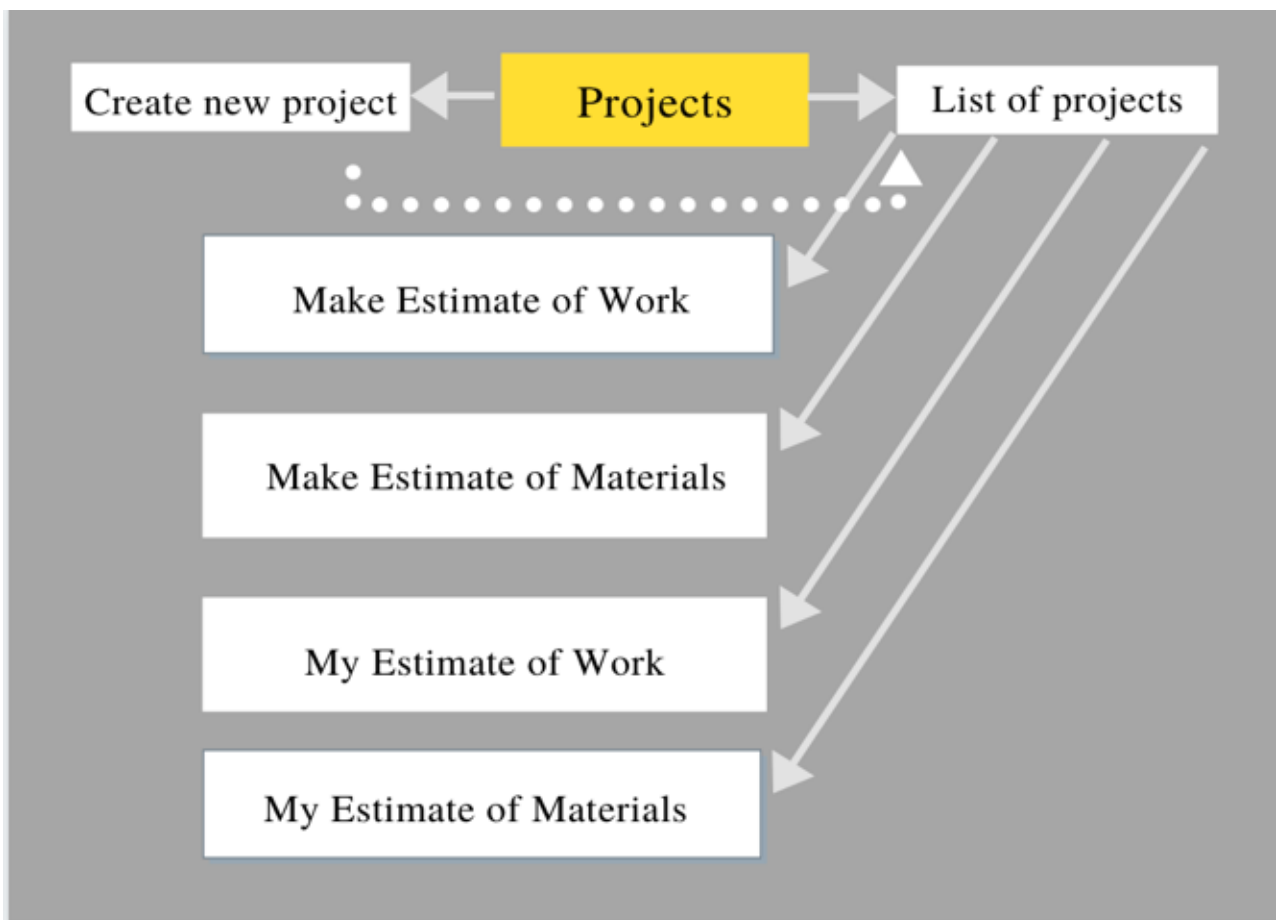


Рисунок 2.4 – Карта блоку Projects

My Customers – на сторінці розміщені всі клієнти. Обравши замовника, відобразиться список проектів, обравши проект – оціночна документація. На даній сторінці відображатимуться статуси кошторисів.

Administration – відповідає за адміністративну систему. У даній секції адміністратор вносить корективи в базу даних. Включає в себе: види робіт (Work examples), матеріали (Materials), одиниці вимірювання (Units), країни (Countries). Адміністратором може виступати тільки один користувач-виконавець. Він може надавати доступ до адміністративного меню у вигляді кода. Даний код можна змінювати в налаштуваннях так само як особисті дані.

Найголовніша частина проекту – створення форми заповнення кошторису будівельних робіт та кошторису на матеріали. Вона має містити в собі:

- найменування кошторису (крім основного створюються додаткові, які виникають під час будівельного процесу);
- функція – максимальна ціна (Max price);
- нумерація;
- додавання рядка (+ Add work);
- можливість видалення рядка;
- види робіт (Work type): зовнішні, внутрішні та робота з фурнітурою;
- опис роботи (Description);
- одиниця вимірювання (Unit): фіксована та введення вручну;
- ціна (Price): фіксована та введення вручну;
- кількість (Quantity);
- загальна сума (Total amount);
- підсумкова ціна (Total price);
- функція збереження кошторису;
- функція завантаження кошторису;
- функція відправлення кошторису в форматі pdf на електронну пошту замовника.

Макет форми розрахунку кошторису відображено на рис. 2.5.

Рисунок 2.5 – Макет форми розрахунку кошторису

Обрання форми web-розробки дозволяє виконавцю створювати кошторис будівельних робіт, перебуваючи на об'єкті, уточнюючи у замовника деталі від великих до малих дрібниць ремонтно-будівельних робіт. Після обговорень клієнт отримує в кабінеті користувача або на електронну пошту потрібну йому інформацію у вигляді кошторису.

Таким чином, як зі сторони клієнта так і адміністратора існує інформаційна потреба. Розглянемо потреби користувачів в табл. 2.1.

Таблиця 2.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Ознайомлення з інформацією про компанію	Клієнт
UN-02	Перегляд фотозвітності компанії	Клієнт
UN-03	Перегляд відгуків користувачів	Клієнт

Продовження таблиці 2.1

ID	Потреби користувача	Джерело
UN-04	Перегляд контактної інформації	Клієнт
UN-05	Можливість створення власного відгуку	Клієнт
UN-06	Можливість погодження кошторису	Адміністратор Клієнт
UN-07	Введення та редагування даних на кошторис	Адміністратор
UN-08	Формування кошторису	Адміністратор
UN-09	Друк кошторисів	Адміністратор Клієнт
UN-10	Систематизування кошторисів	Адміністратор Клієнт
UN-11	Ознайомлення з цінами на матеріали, з використанням пошукової системи та фільтрів	Клієнт
UN-12	Ознайомлення з кошторисами	Клієнт
UN-13	Відправлення кошторису на електронну пошту	Адміністратор
UN-14	Обмежений доступ до адміністративних функцій	Адміністратор
UN-15	Формування кошторису в pdf форматі	Адміністратор Клієнт
UN-16	Надійність паролю	Адміністратор Клієнт

Визначимо та розподілимо системні вимоги та відобразимо в табл.2.2.

Таблиця 2.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність сторінки з інформаційним контентом	М	Відповідає за заповнення web-сайта контентом

## Продовження таблиці 2.2

ID	Системні вимоги	Пріоритет	Опис
SR-02	Функція відображення видів робіт	М	Відповідає за відображення видів робіт у формі при формуванні кошторису
SR-03	Функція відображення одиниці вимірювань	М	Відповідає за відображення одиниці вимірювань у формі при формуванні кошторису
SR-04	Функція відображення типів робіт	М	Відповідає за відображення типів робіт у формі при формуванні кошторису
SR-05	Функції відображення цін на послуги	М	Надає можливість подальшого відображення цін на всі види робіт
SR-06	Функції відображення цін на матеріал	М	Надає можливість подальшого відображення цін на всі види матеріалів
SR-07	Функції відображення клієнтів	М	Відповідає за відображення списку клієнтів в аккаунті адміністратора
SR-08	Функція шифрованих паролей	М	Відповідає за конфіденціальність
SR-09	Функції відображення кошторисів по роботі	М	Відповідає за відображення кошторисів по роботі
SR-10	Функції відображення кошторисів по матеріалам	М	Відповідає за відображення кошторисів по матеріалом

## Продовження таблиці 2.2

ID	Системні вимоги	Пріоритет	Опис
SR-11	Функції відображення проектів	M	Відповідають за відображення проектів у системі
SR-12	Функції відображення відгуків	S	Формує можливість переглядів відгуків
SR-13	Панель адміністратора	M	Відповідає за подальше наповнення та редагування основних інформаційних довідників адміністратором

Умовні позначення для пріоритету, які використовуються в табл. 2.2:

Must have (M) – першочергові вимоги для реалізації;

Should have (S) – другорядні вимоги, які повинні бути виконані;

Could have (C) – додаткові вимоги.

ПЗ повинне задовольняти наступні веб-браузери: Safari 3.2.1 і вище, Internet Explorer 7.0 і вище, Firefox 3.5 і вище, Opera 9.5 і вище, Chrome 2 і вище та адаптований під мобільні версії. Включена підтримка javascript, Flash і cookies.

Було проведено планування розробки веб-сайту. Розроблена ієрархічна структуру робіт, побудована діаграма Ганта, проведені аналіз та оцінки ризиків та сплановані заходи їх запобігання (Додаток А).

## 2.2 Вибір засобів реалізації веб-орієнтований інформаційної системи формування кошторисів будівельних робіт

Розробка веб-орієнтованої інформаційної системи формування кошторисів будівельних робіт складатиметься з Front-end (HTML, CSS, JavaScript, Vue.js), Back-end (PHP, Laravel Framework) та розробки бази даних (MYSQL).

Vue.js – це JavaScript-фреймворк, який підходить для створення високоадаптованих користувацьких інтерфейсів і складних односторінкових додатків [28]. Основна перевага це простота інтеграції з іншими проектами та бібліотеками (так як він заснований на JavaScript і ядро орієнтоване на задачі рівня представлення, що спрощує об'єднання з іншими бібліотеками), в тому числі і з Back-end фреймворками. Він легкий у використанні. Vue.js орієнтований на більш класичну модель розробки (HTML, CSS, JS). Для побудови веб-інтерфейсів використовуються шаблони архітектури MVVM (Model-View-ViewModel) [29].

Основні функції: реактивні інтерфейси, декларативний рендеринг, зв'язування даних, директиви (всі директиви мають префікс «V-»). В директиву передається значення стану, а в якості аргументів використовуються html атрибути або VueJS події), логіка шаблонів, компоненти, обробка подій, властивості, переходи і анімація CSS; фільтри [30].

Vue.js використовують такі компанії, як Xiaomi, Alibaba, WizzAir, EuroNews, Grammarly, Gitlab и Laracasts, Adobe, Behance, Codeship, Reuters. Це говорить про те, що продукт є популярним та має свої переваги. До них відносяться: посилений HTML, детальна документація (document object model – DOM), адаптивність (має схожість з Angular та React по дизайну та архітектурі), інтеграція, масштабування і малі розміри, що гарантує



мінімальне навантаження на проект та швидке завантаження сайту [31]. Vue.js найкраще підходить для розробки рішень, які використовують зовнішні API для обробки даних. Тобто Vue.js гнучкий у роботі з шаблоном, має простий синтаксис і низький поріг входження, швидкий рендеринг і незначні розміри, використовуються розширення для браузера DevTool та прогресивну адаптивність [32].

Для написання Back-end використовуватиметься мова PHP, фреймворк Laravel та MySQL.

PHP – одна з найпопулярніших мов, яка використовується в процесі розробки Back-end сервера. Існує багато фреймворків, які побудовані на PHP. Фреймворки надають структуру для створення додатків та містять безліч допоміжних функцій, які допомагають процесу розробки програми. Найбільш популярними є: Laravel, CodeIgniter, Phalcon, Zend, Symfony, Yii, CakePHP, FuelPHP та інші [33]. Всі вони є подібними та мають різний підхід в розробленні.

Використання фреймворків має багато переваг:

- Прискорення процесу розробки;
- Спрощення всього процесу;
- Спрощення обслуговування додатків;
- Забезпечення функції високого захисту;
- Автоматизація основних завдань у розробці;
- Ефективне використання баз даних [34].

Laravel – це веб-фреймворк з відкритим кодом PHP. Його архітектурні зразки в основному базуються на Symfony [35].

За статистикою по використанню фреймворків Laravel займає перше місце, перевищуючи кількість користувачів вдвічі [33].

Цей фреймворк набув більшої популярності після включення в нього таких функцій, як інтерфейс командного рядка під назвою Artisan, підтримка системи баз даних та міграції.

Розглянемо основні переваги Laravel:

- Високий рівень безпеки. Кількість кібератак зростає з кожним днем, тому Laravel забезпечує безпечне середовище, захищаючи веб-програму від таких проблем, як SQL Injection та CSRF (Підробка міжсайтових запитів);
- Аутентифікація. Laravel пропонує готову конфігурацію для системи автентифікації та авторизації;
- Відкритий код. Це дозволяє будь-кому прийняти участь в модернізації фреймворку та його сторонніх додатків;
- Підтримка MVC та об'єктно-орієнтований підхід. Laravel – це суто MVC-сумісний фреймворк, який забезпечує вбудовану підтримку принципу IOC (Inversion of Control). Model View Controller допомагає легко змінювати і модифікувати шаблон та коди;
- Artisan. Ця функція звільняє розробника від створення належних скелетів коду;
- Eloquent ORM. Laravel має найкращий об'єктно-реляційний Mapper порівняно з іншими фреймворками. Це об'єктно-реляційне відображення дозволяє взаємодіяти з об'єктами бази даних та зв'язками баз даних за допомогою виразного синтаксису.
- Шаблонізатор Blade. За допомогою цього механізму можна стандартизувати та багаторазово використовувати один і той же шаблон в різних частинах додатку.
- Планування завдань;
- Тестування;
- Міграція баз даних. Можна легко змінити структуру бази даних;
- Інтеграція з mail-сервісом. Laravel надає підтримку відправленню повідомлень через безліч каналів доставки, включаючи SMS і Slack;
- Швидкість розробки;
- Багатомовність;

– Об'єктно-орієнтовані бібліотеки. Всі бібліотеки в Laravel взаємодіють легко. Об'єктна орієнтованість дозволяє повторно використовувати код та уникнути його дублювання [34, 36].

Недоліками фреймворку є офіційна документація англійською мовою, яку розуміють не всі, порушений зворотній зв'язок між різними версіями Laravel, каталоги та файли не завжди логічно розташовані. Порівнюючи з фреймворком Yii в Laravel відсутній вбудований генератор сторінок для перегляду, редагування, видалення записів з баз даних. Порівнюючи з такими фреймворками, як Ruby on Rails та Django, Laravel має обмежену вбудовану підтримку із-за не великої ваги, але це вирішується за допомогою сторонніх інструментів [37].

Щодо недоліків такого популярного фреймворка, як Symfony, то він надає неповну реалізацію моделі MVC, а лише модель та контролер. Фреймворк є достатньо важким для засвоєння. Розглядаючи фреймворк Yii, то його недоліками можна назвати невирішене питання повторення коду, сильний зв'язок класів та слабка інтеграція шаблонізатора (Twig, Smarty).

Фреймворк Laravel – це сучасний, логічний, продуманий, зрозумілий інструмент для розробки веб-сайтів. Включає в себе найкращий набір інструментів із попередньо встановленими об'єктно-орієнтованими бібліотеками. Такі функції, як моніторинг активних користувачів, шифрування та захист CSRF, роблять його потужнішим, ніж інші платформи.

У роботі використаємо систему управління базами даних MySQL, яка є гарним рішенням для невеликих розробок. MySQL має високу швидкість роботи, оптимальну надійність та швидку обробку даних. Вона являє собою ПЗ з відкритим доступом та є безкоштовною.

На протязі розробки інформаційної системи потрібно проводити функціональне тестування. Краще за все його проводити на локальному веб-сервері. Тому для Mac OS існує додаток MAMP (Macintosh, Apache, MySQL та

PHP). Apache – це крос-платформне програмне забезпечення веб-сервера з відкритим кодом та який підтримує протокол HTTP.

Середовищем розробки візьмемо PhpStorm. Середовище має наступні можливості: редактор PHP, HTML, CSS та JavaScript, навігація по коду, автодоповнення кодів, підтримка утиліт, можливість установки плагінів, інструмент роботи з базою даних та інші. PhpStorm має плагін The Laravel IDE Helper, який допомагає в розробці додатків, роблячи підказки при написанні коду.

## **3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФОРМУВАННЯ КОШТОРИСУ БУДІВЕЛЬНИХ РОБІТ**

### **3.1 Структурно-функціональне моделювання формування кошторису будівельних робіт**

Найбільш популярною методологією функціонального моделювання є IDEF0. Вона використовується для створення функціональної моделі, яка є структурованим зображенням функцій виробничої системи, середовища, інформації і об'єктів, що зв'язують ці функції [38].

Структурно-функціональне моделювання веб-сайту формування кошторису будівельних робіт зображується у діаграмі IDEF0 (рис. 3.1). Контекстна діаграма показує призначення системи та її взаємодію із зовнішнім середовищем. Кожна модель має лише одну контекстну діаграма [39], яка складається з функціонального блоку, який відповідає за конкретний процес та зображується в прямокутнику, а також стрілок, що входять в цей блок з кожної сторони. Головна функція веб-сайту – формування кошторису будівельних робіт.

Об'єкти, які впливають на процес, зображаються стрілками:

- “Вхід” (ліва сторона): запит на замовлення будівельних робіт, види робіт та їх вартість, дані про фінансові можливості замовника;
- “Управління” (верхня сторона): шаблон кошторису, порядок оформлення кошторисної документації;
- “Механізми” (нижня сторона): адміністратор, замовник, Веб-орієнтована інформаційна система, web-сервер, апаратне забезпечення;
- “Вихід” (права сторона): кошторис у базі даних, pdf файл кошторису.



Рисунок 3.1 – Контекстна діаграма

Після опису головної функції в контекстній діаграмі виконується її декомпозиція. На даному етапі важливо визначити функції, які впливають на основну. Далі функції можуть ділитися на підфункції поки не досягнеться потрібний рівень деталізації аналізуючої системи. Діаграми, які описують кожну підфункцію системи, називаються дочірніми діаграмами декомпозиції [39].

У декомпозиції моделі головної функції веб-сайту визначено п'ять блоків:

- створення проекту кошторису;
- заповнення форми кошторису;
- перегляд замовлення;
- підтвердження кошторису;
- формування документації.

Як і в контекстній діаграмі, стрілки мають те ж саме значення, але додаються нові, які виходячи з першого блоку входять в інший, що деталізує функціонування веб-сайту.

Діаграма декомпозиції зображена на рис. 3.2.

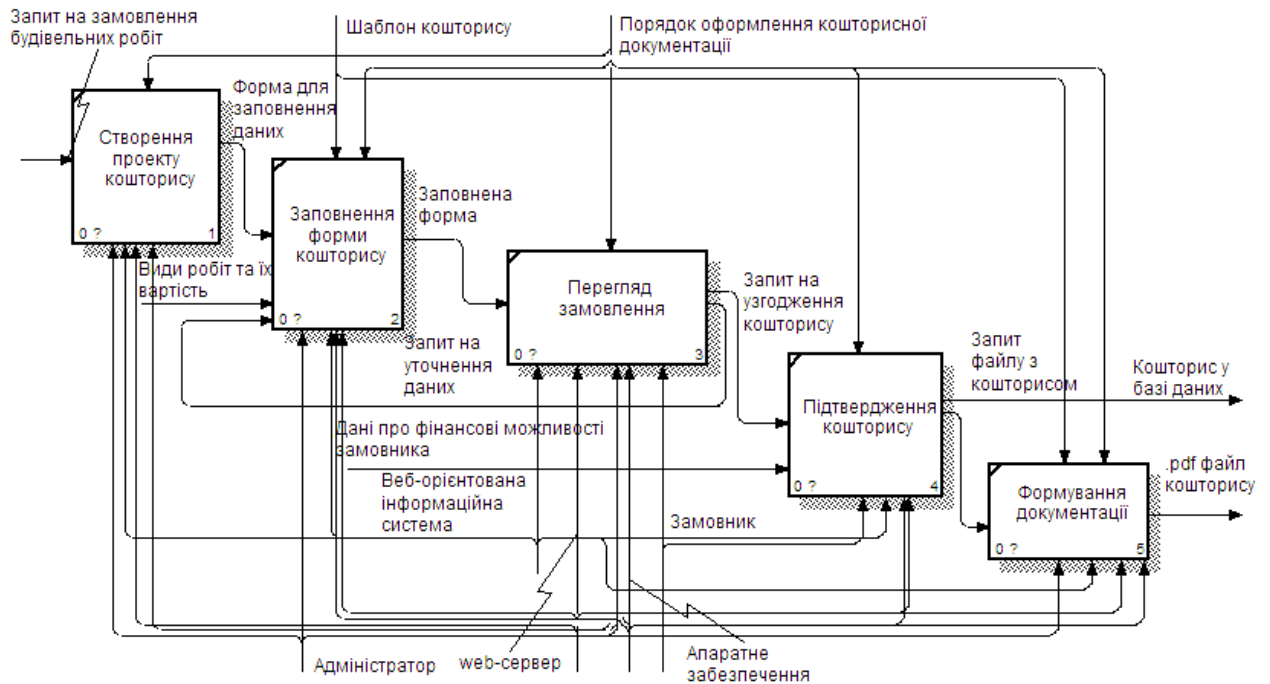


Рисунок 3.2 – Діаграма декомпозиції

Блок “Створення проекту кошторису”:

- Вхідною стрілкою є “Запит на замовлення будівельних робіт”;
- Вихідною стрілкою – “Форма для заповнення даних”;
- Стрілка контролю – “Порядок оформлення кошторисної документації”;
- Стрілки механізму – “Адміністратор”, “Веб-орієнтована інформаційна система”, “web-сервер”, “Апаратне забезпечення”.

Блок “Заповнення форми кошторису”:

- Вхідними стрілками є “Форма для заповнення даних”, “Види робіт та їх вартість”, “Запит на уточнення даних”;
- Вихідною стрілкою – “Заповнена форма”;
- Стрілки контролю – “Шаблон кошторису”, “Порядок оформлення кошторисної документації”;

- Стрілки механізму – “Адміністратор”, “Веб-орієнтована інформаційна система”, “web-сервер”, “Апаратне забезпечення”.

Блок “Перегляд замовлення”:

- Вхідною стрілкою до блоку є “Заповнена форма”;
- Вихідними стрілками – “Запит на узгодження кошторису”, “Запит на уточнення даних”;
- Стрілка контролю – “Порядок оформлення кошторисної документації”;
- Стрілки механізму – “Адміністратор”, “Замовник”, “Веб-орієнтована інформаційна система”, “web-сервер”, “Апаратне забезпечення”.

Блок “Підтвердження кошторису”:

- Вхідними стрілками є “Запит на узгодження кошторису”, “Дані про фінансові можливості замовника”;
- Вихідними стрілками – “Запит файлу з кошторисом”, “Кошторис у базі даних”;
- Стрілка контролю – “Порядок оформлення кошторисної документації”;
- Стрілки механізму – “Замовник”, “Веб-орієнтована інформаційна система”, “web-сервер”, “Апаратне забезпечення”.

Блок “Формування документації”:

- Вхідною стрілкою до блоку є “Запит файлу з кошторисом”;
- Вихідними стрілками – “pdf файл кошторису”;
- Стрілка контролю – “Шаблон кошторису”, “Порядок оформлення кошторисної документації”;
- Стрілки механізму – “Адміністратор”, “Веб-орієнтована інформаційна система”, “web-сервер”, “Апаратне забезпечення”.



### **3.2 Моделювання варіантів використання web-орієнтованої інформаційної системи формування кошторису будівельних**

Варіант використання (VV) застосовується для вираження потрібної поведінки системи, яку розробляють, не описуючи її реалізацію. Це дозволить бути більш впевненим в архітектурних рішеннях і перевірити систему в ході її розроблення [40].

Діаграми VV (Use Case) є основним видом діаграм при моделюванні поведінки системи. Кожна з них показує набір VV і акторів у їхній взаємодії [40]. UML діаграми VV використовуються для візуалізації поведінки системи, щоб користувач зміг зрозуміти, як використовувати той чи інший елемент, а розробник – як його реалізувати[41].

Актор – це будь-яка сутність, яка виконує роль впливу на систему, моделюючи розробником. Варіант використання виконують функцію опису сервісів, які надаються актору системою[41].

Взаємодія акторів з веб-сайтом зображено на рис. 3.3

Актори:

Адміністратор – особа, яка має доступ до адміністративної частини веб-сайту, має право коригувати бази даних, приймати заявки, формувати кошториси будівельних робіт і на матеріали.

Клієнт – особа, яка використовує веб-сайт, ознайомлюючись з інформацією, має можливість зареєструватися, здійснити запит, залишити відгук.

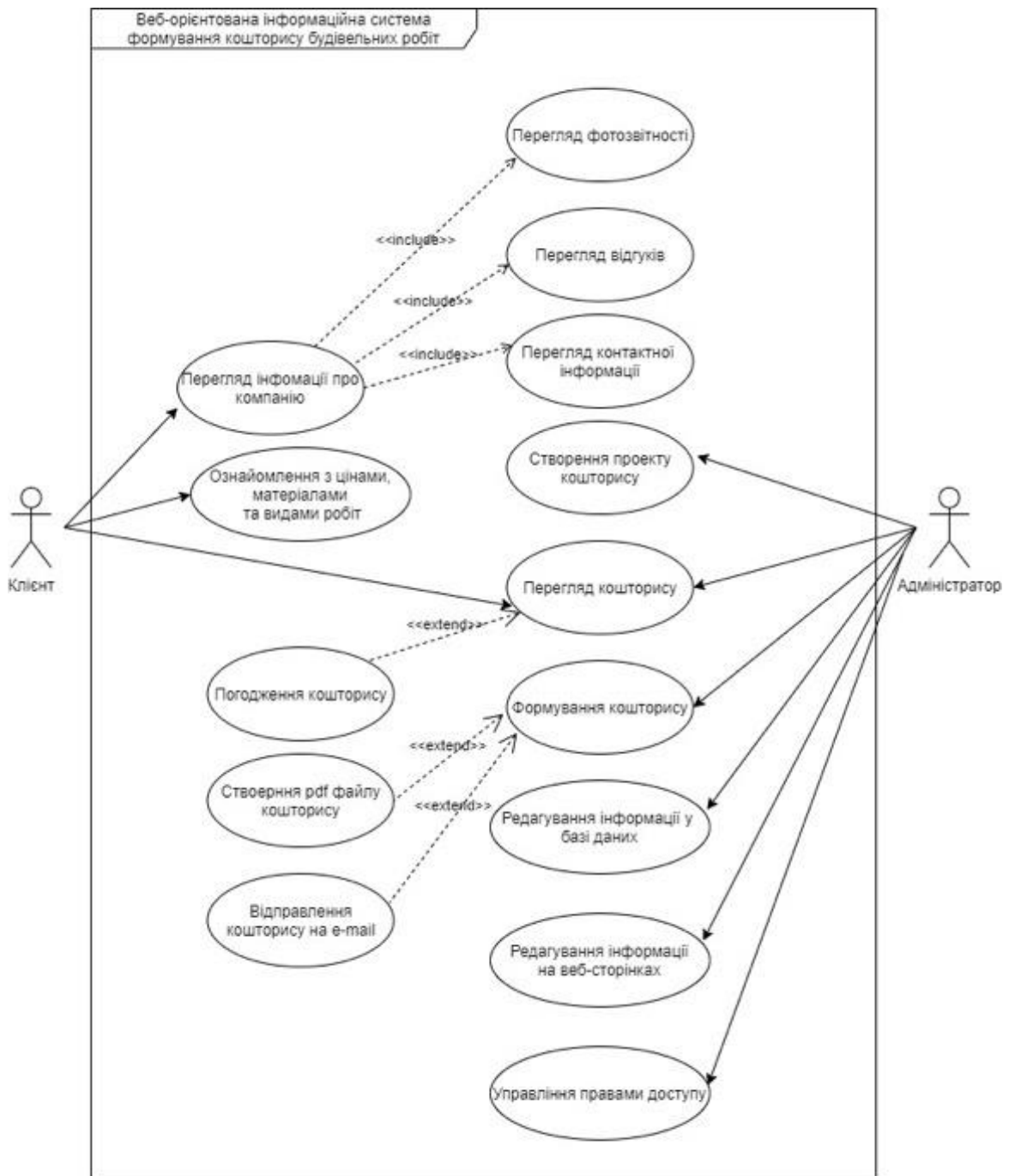


Рисунок 3.3 – Діаграма варіантів використання

Варіанти використання:

- ВВ Перегляд інформації про компанію – надає дозвіл клієнту ознайомлюватися з інформацією про компанію;
- ВВ Перегляд фотозвітності – надає дозвіл клієнту познайомитися з роботами компанії;
- ВВ Перегляд відгуків – надає дозвіл переглянути відгуки клієнтів, які вже скористалися послугами даної компанії;

- ВВ Перегляд контактної інформації – надає дозвіл отримати контактну інформацію компанії;
- ВВ Ознайомлення з цінами, матеріалами та видами робіт – надає дозвіл клієнту після авторизації познайомитися з цінами на матеріали та роботи;
- ВВ Створення проекту кошторису – надає дозвіл створення проекту кошторису;
- ВВ Перегляд кошторису – надає дозвіл переглянути кошторис робіт та матеріали;
- ВВ Погодження кошторису – надає можливість погодити кошториси з клієнтом з допомогою веб-сайту;
- ВВ Формування кошторису – надає можливість сформувати кошториси;
- ВВ Створення pdf файлу кошторису – надає можливість створення pdf файлу кошторису після збереження його в базі даних;
- ВВ Відправлення кошторису на e-mail – надає можливість адміністратору відправити кошторис на e-mail клієнту з особистого аккаунту, відразу після його формування;
- ВВ Редагування інформації у базі даних – надає можливість змінити інформацію у базі даних, яка стосується описання видів робіт, цін на види роботи та матеріали та ін.;
- ВВ Редагування інформації на веб-сторінках – надає дозвіл редагувати інформації на веб-сайті;
- ВВ Управління правами доступу – надає дозвіл зареєструватися як клієнт або адміністратор; кожний аккаунт має своє функціональне забезпечення; кожен пароль є зашифрованим.

### 3.3 Проектування моделі бази даних

База даних – це сховище інформації, яке є систематизованим та централізованим, певної предметної області та до якої мають доступ різні користувачі. Основне завдання – збереження великих обсягів інформації та надання доступу до неї прикладній програмі або користувачу [42].

Розглянемо реляційну базу для проектування моделі бази даних. Реляційні бази даних розроблені для швидкого збереження та отримання великих об'ємів інформації. Дана база представляє собою набір таблиць (сутності). Таблиця складається із колонок (атрибути) та рядків (кортежі). Рядки в свою чергу складаються із значень колонок. Всі рядки однієї таблиці мають одну структуру. Всередині таблиць можуть бути визначені обмеження, між таблицями існують відносини [43].

Реляційна база даних дає можливість швидко порівняти інформацію, що дає можливість створення нових таблиць із необхідним формацією, створюючи зв'язки з існуючими таблицями. Це підвищує швидкість бази даних [42].

Розглянемо таблиці бази даних в табл. 3.1.

Таблиця 3.1 – Призначення таблиць бази даних

№ з/п	Назва таблиці	Призначення
1	Constraction projects	Збереження інформації проекту.
2	Constraction estimate_work	Створює та зберігає інформацію по кошторису робіт. Виступає сполучною ланкою, до якої кріпляться інші, як результат на цій базі формуватиметься кошторис.
3	Constraction example_works	Інформація по прикладам/видам робіт, яку редагує адміністратор.

## Продовження таблиці 3.1

№ з/п	Назва таблиці	Призначення
4	Constraction works	Єднальна таблиця, яка відповідає за додавання строки в форму для створення кошторису, заповнюючи інформацію по роботах (тип роботи, приклад роботи, одиниці вимірювання, вартість за 1 одиницю вимірювання, кількість, загальна сума по відповідному прикладу роботи).
5	Constraction type_work	Інформація по типам робіт для заповнення кошторису.
6	Constraction estimate_materials	Створює та зберігає інформацію по кошторису матеріалів. Виступає сполучною ланкою, до якої кріпляться інші, як результат на цій базі формуватиметься кошторис.
7	Constraction materials	Єднальна таблиця, яка відповідає за додавання строки в форму для створення кошторису, заповнюючи інформацію по матеріалам (вибір матеріалу, вартість за 1 одиницю, кількість, загальна сума по відповідному матеріалу).
8	Constraction materials_for_estimate	Відповідає за наповненням матеріалів для формування кошторису по матеріалам.
9	Constraction units	Інформація про одиниці вимірювання, які можна редагувати.
10	Constraction users	Збереження інформації про зареєстрованих всіх користувачів.
11	Constraction comments	Збереження інформації про відгуки (хто та що написав).

Продовження таблиці 3.1

№ з/п	Назва таблиці	Призначення
12	Constraction countries	Інформація у вигляді списку країн для заповнення анкети.
13	Constraction migration (Системна таблиця)	Інформація для системної роботи фреймворку Laravel. Збереження файлів, які зробили міграцію
14	Constraction password_resets	Таблиця для скидання паролей

Для кращого розуміння бази даних використовують ER-діаграму, яка відображає ключові сутності, атрибути та зв'язки між сутностями. На рис. 3.4 зображено ER-діаграму створеної бази даних [44]

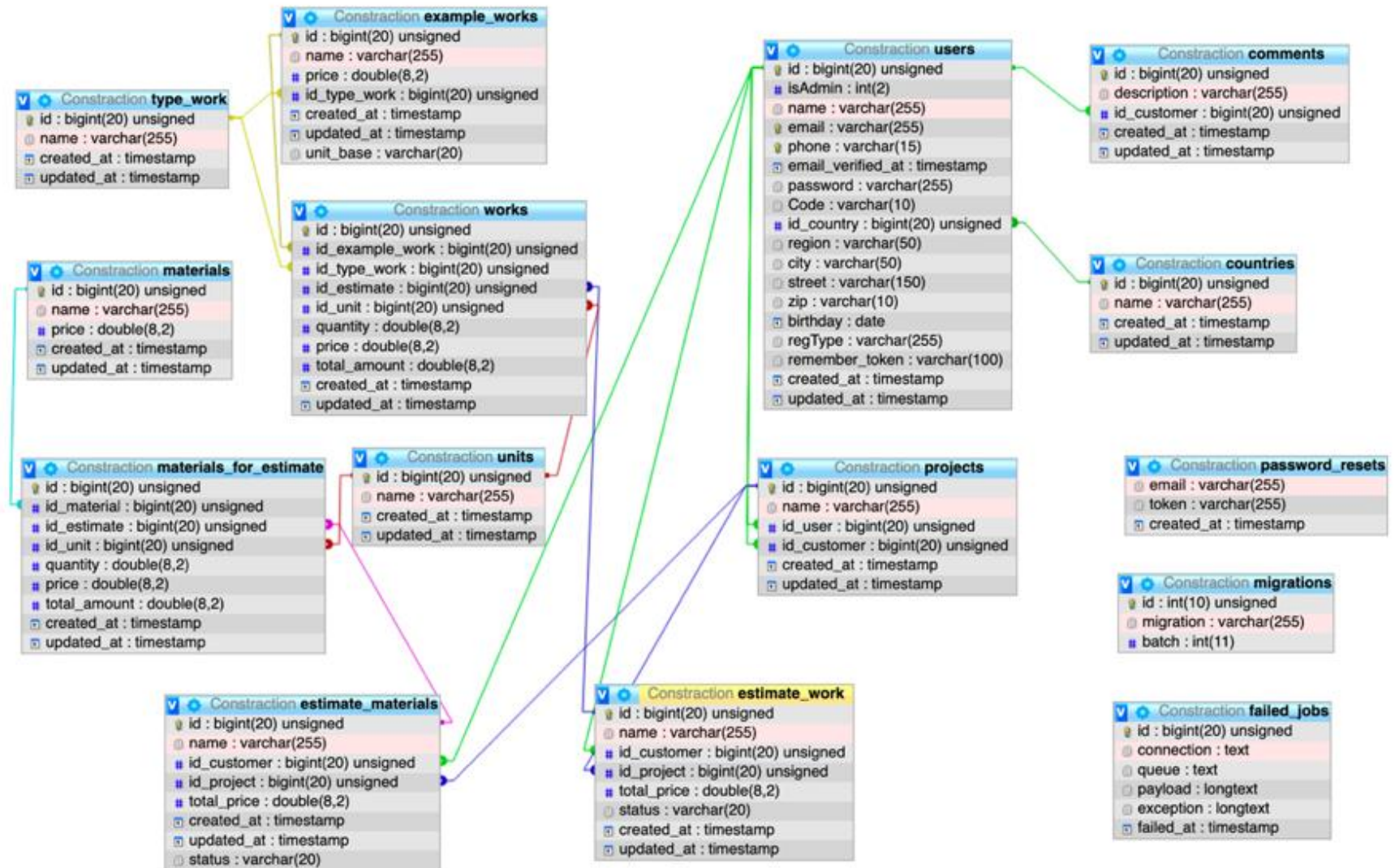


Рисунок 3.4 – ER-діаграма бази даних

## 4 РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФОРМУВАННЯ КОШТОРИСУ БУДІВЕЛЬНИХ РОБІТ

### 4.1 Розробка архітектури інформаційної системи

Архітектура web-сайту робить систему зрозумілішою для користувача, легкою у знаходженні інформації та здійсненні переходів.

Було вирішено розробити веб-сайт в трірівневій архітектурі на основі моделі «Model-View-Controller» (MVC).

Трирівнева архітектура містить в собі три компоненти: клієнт (браузер, через який користувач переглядає веб-сайт), сервери (на якому знаходиться наш веб-сайт, використання фреймворку Laravel) та сервер баз даних, які використовуються сайтом (MySQL).

Систему реалізовано за допомогою php-фреймворку Laravel. Шаблон MVC, на якому фреймворк побудований, застосовується для відокремлення даних (Model) від інтерфейсу користувача (View) так, щоб зміни інтерфейсу користувача мали мінімальний вплив на роботу з даними, а зміни в моделі даних могли не несуть змін в інтерфейсі користувача [45].

Модель відповідає за зберігання, обчислення, формування, отримання результатів. А також відбувається формування результату обчислень, стрічок, масивів, об'єктів тощо.

View (Вигляд) призначений для показу результатів на екран користувача. Як правило, це HTML-розмітка в браузері.

Контролер є свого роду координатором між моделлю та виглядом. Контролює занесення даних користувачем та використовує модель і вигляд для реалізації необхідної реакції [45].

На рис. 4.1 представлена архітектура веб-сайту, яка розроблено за допомогою php-фреймворку Laravel [46].



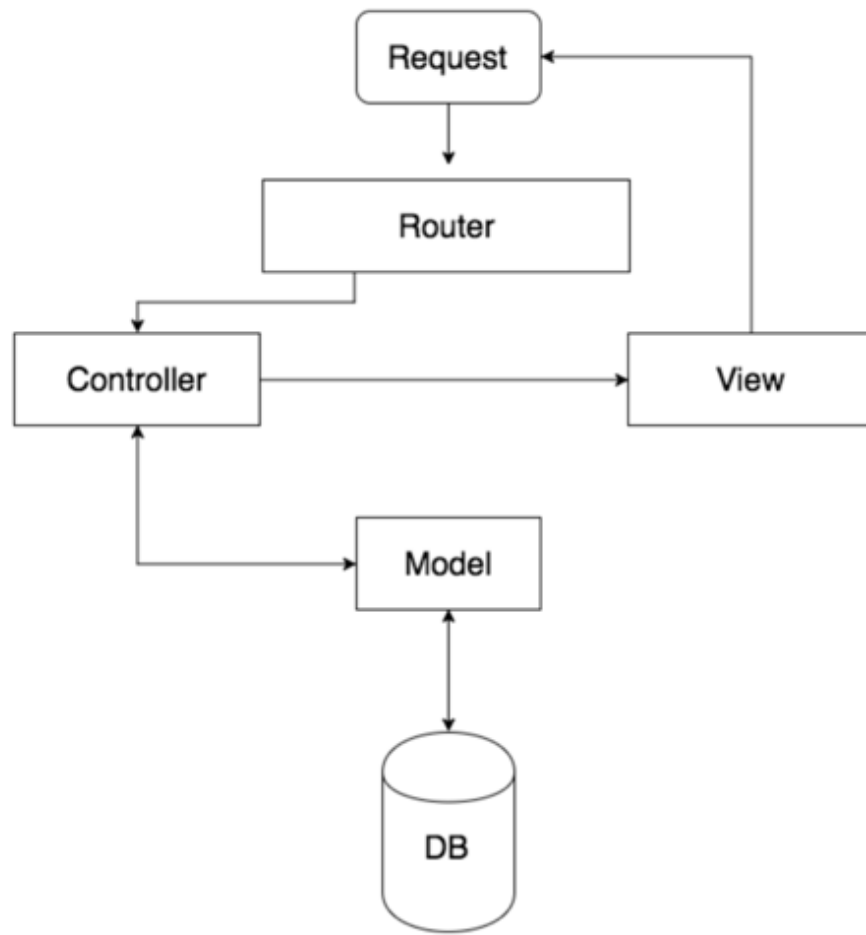


Рисунок 4.1 – Архітектура веб-сайту з використанням php-фреймворку Laravel

При взаємодії з веб-сайтом клієнт відправляє запит (URL-адресу), веб-сервер отримує його та передає до компонента Laravel, який відповідає за маршрутизацію. У свою чергу, маршрутизатор (Router) приймає запит і перенаправляє його на обробку у відповідний метод класу контролера. У середині методу реалізований алгоритм, який здійснює взаємодію з моделями. Моделі відповідають таблицям-сутностям бази даних. Взаємодія може відбуватися наприклад через драйвер (ця частина коду яка реалізує підключення і взаємодію з базою даних). Отже, контролер взаємодіє з моделлю, генерує відповідь у вигляді “View”, використовуючи HTML, CSS та зображення. Потім повертає веб-сторінку в браузер користувача [46].

Використання фреймворку Laravel забезпечує слідування архітектурному шаблону MVC, а значить і гнучкість та масштабування сайту.

## 4.2 Встановлення та розробка компонентів інформаційної системи

Для розробки проекту було виконано ряд дій. На рис. 4.2 відображена послідовність розробки проекту.

○ При добавлении новых estimate выводить меню проекта	2020-08-18 16:12
○ Оптимизация функций для php 7.3	2020-08-18 15:35
○ Сортировка комментариев по убыванию	2020-08-03 00:45
○ Изменен внешний вид страниц	2020-08-03 00:41
○ Исправлено удаление работ в режиме редактирования	2020-08-02 23:59
○ Реализован вывод цен на материалы	2020-08-02 23:54
○ Реализован функционал feedback	2020-08-02 17:57
○ Реализован функционал отправки PDF на Email	2020-08-01 15:15
○ Реализован функционал просмотра текущих проектов клиентом с возм формирования pdf	2020-08-01 12:34
○ Добавлена колонка заказчиков в таблице проектов	2020-08-01 11:49
○ Реализован функционал просмотра заказчиков, просмотра созданных на него всех estimates, формирование pdf,...	2020-07-30 23:58
○ Реализован функционал редактирования профиля пользователя	2020-07-28 22:38
○ Реализован функционал создания/просмотра/редактирования/удаления Estimate of materials + формирование PDF	2020-07-26 22:36
○ Реализован компонент select с поиском + добавлена кнопка загрузки цены материала	2020-07-25 22:55
○ Реализовано отображение формы создания Estimate material(настройка роутов, создание шаблонов)	2020-07-25 13:20
○ созданы миграции бд под estimate material	2020-07-25 13:18
○ Доработка таблицы PDF	2020-07-22 19:32
○ Переработан функционал выбора работ под новую структуру	2020-07-21 23:59
○ Реализован функционал формирования PDF Estimate Work	2020-07-19 14:28
○ Реализован функционал просмотра/редактирования/удаления Estimate Work	2020-07-18 14:43
○ Реализован функционал проверки максимальной стоимости	2020-07-17 23:48
○ Настройка связей БД	2020-07-17 22:14
○ Реализован интерфейс создания Estimate	2020-07-14 21:34
○ Добавлена привязка проекта к пользователю	2020-07-09 22:57
○ Реализация создания, ред, удаления проектов	2020-07-07 00:06
○ Добавлены страницы "О нас", "Галерея", переработана главная страница	2020-07-05 14:28
○ Реализована система регистрации и авторизации + смена пароля	2020-07-02 22:58
○ Добавлено изображение главной страницы	2020-06-29 12:14
○ Установка модуля авторизации + Vue	2020-06-29 11:40
○ Установка Laravel 7	2020-06-29 11:18

Рисунок 4.2 – Послідовність розробки проекту

Розпочнемо опис процесу розробки з бази даних, так як існує постійна взаємодія з нею. На рис. 4.3 показана база даних Construction. Вона вміщує в собі наступні таблиці: comments, countries, estimate\_materials, estimate\_work, example\_works, failed\_job, materials, materials\_for\_estimate, migrations, password\_resets, projects, type\_work, units, users, works.

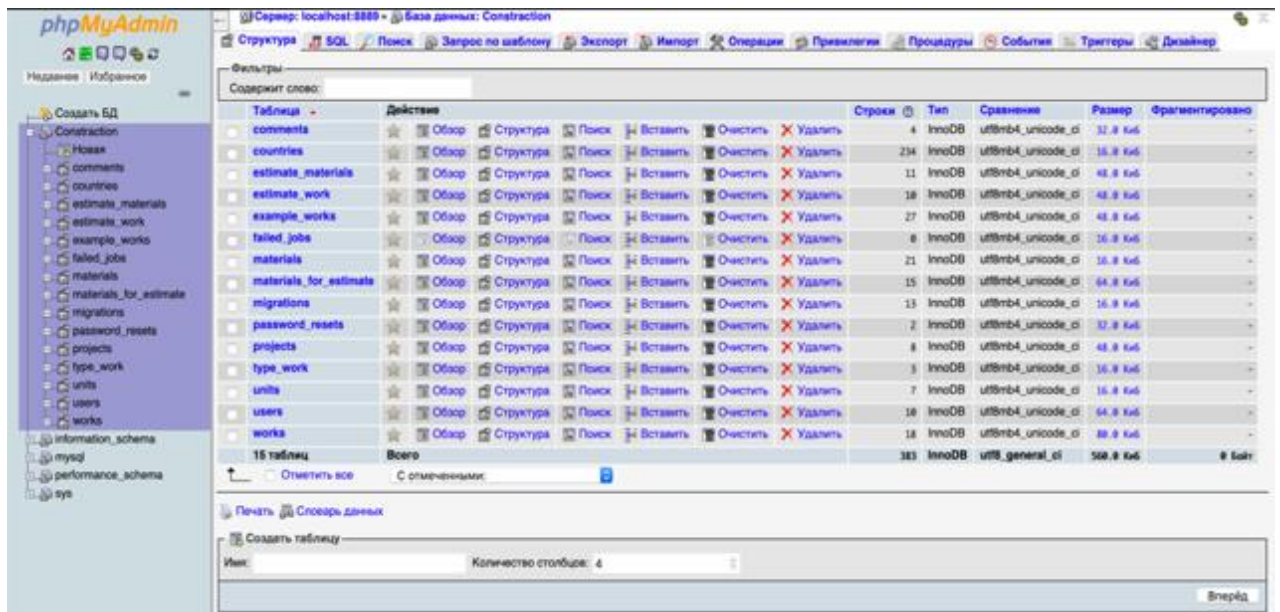


Рисунок 4.3 – База даних у phpMyAdmin

Одним із перших важливих кроків у розробці проекту став крок створення моделей, які відповідає за взаємодію з базою даних (рис. 4.4). У моделях вказується найменування бази даних, первинний ключ та fillable (поля, які можуть бути доступними при масових заповненнях). Моделі являються посиланнями до таблиць (для спрощення запиту бази даних). Через модель можна вибрати, який саме рядок потрібно видаляти, додати чи коригувати.

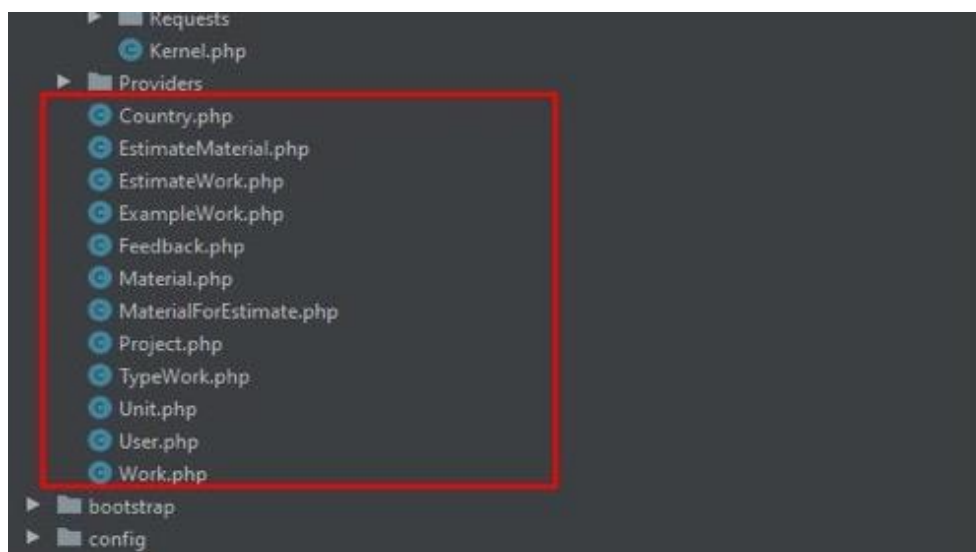


Рисунок 4.4 – Моделі проекту

У розробці було використано міграції. Міграції потрібні для системної роботи Laravel. Міграція – це своєрідний механізм контролю, який дозволяє вирішити проблему неузгодження баз даних. Для цього потрібно написати команду artisan (рис. 4.5).

```

D:\OSPanel\domains\CM>php artisan migrate
Migrating: 2020_07_23_164936_create_materials_table
Migrated: 2020_07_23_164936_create_materials_table (0.66 seconds)
Migrating: 2020_07_23_170516_create_estimate_materials_table
Migrated: 2020_07_23_170516_create_estimate_materials_table (1.78 seconds)
Migrating: 2020_07_23_170859_create_materials_for_estimate_table
Migrated: 2020_07_23_170859_create_materials_for_estimate_table (2.55 seconds)
D:\OSPanel\domains\CM>

```

Рисунок 4.5 – Команда artisan для міграцій

Кожна міграція реалізується як клас та має 2 методи: `up()` – при запуску міграції та `down()` – при відкаті.

На рис. 4.6 знаходяться файли міграції, а на рис. 4.7 бачимо, яку структуру має кожен із файлів міграції. Розглянемо приклад: є функція `Up`, вона добавляє в таблицю поля `id`, тип даних і так далі.

Можна створювати таблицю вручну, але можна й через команду файл міграції. У файлі вказуємо, яка таблиця створюється, всі поля (`name`, `id...`) і потім приписується команда `php artisan migration`. І виходить, що всі міграції, які є в файлі ідуть в базу даних. Якщо в таблиці міграції немає цього файлу, то робиться міграція. Тобто створюється таблиця з тими полями, які прописані у файлі.

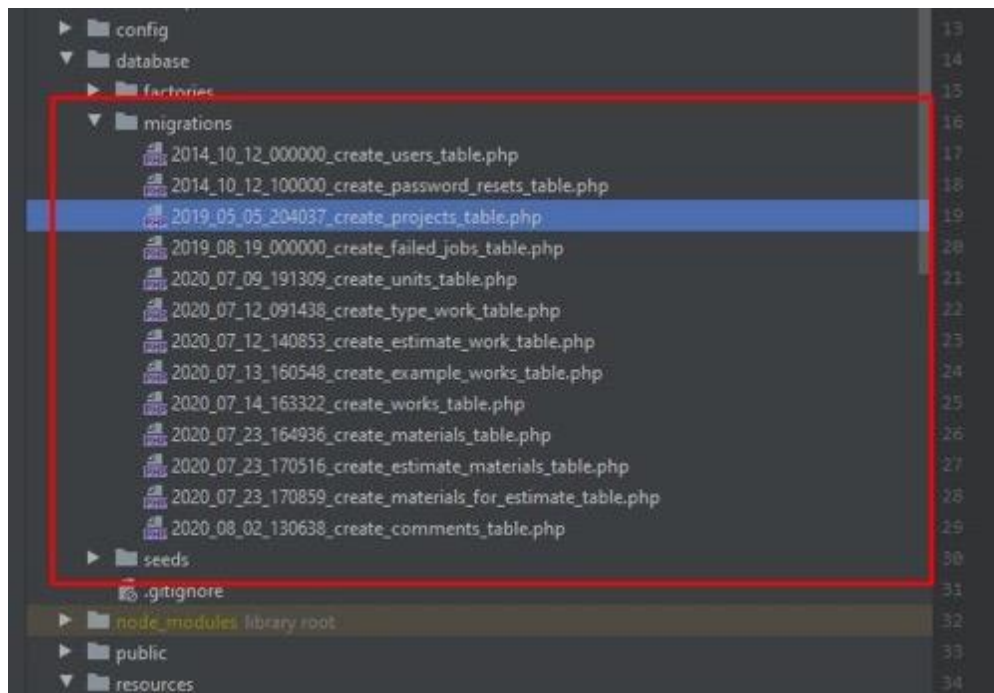


Рисунок 4.6 – Файли міграції



Рисунок 4.7 – Структура міграції

Наступним важливим кроком є створення контролеру (рис. 4.8). У контролерах звертаємося до бази даних через моделі. Вся серверна логіка знаходиться саме в них (збереження, оновлення, видалення, формування pdf, завантаження сторінок та ін.)

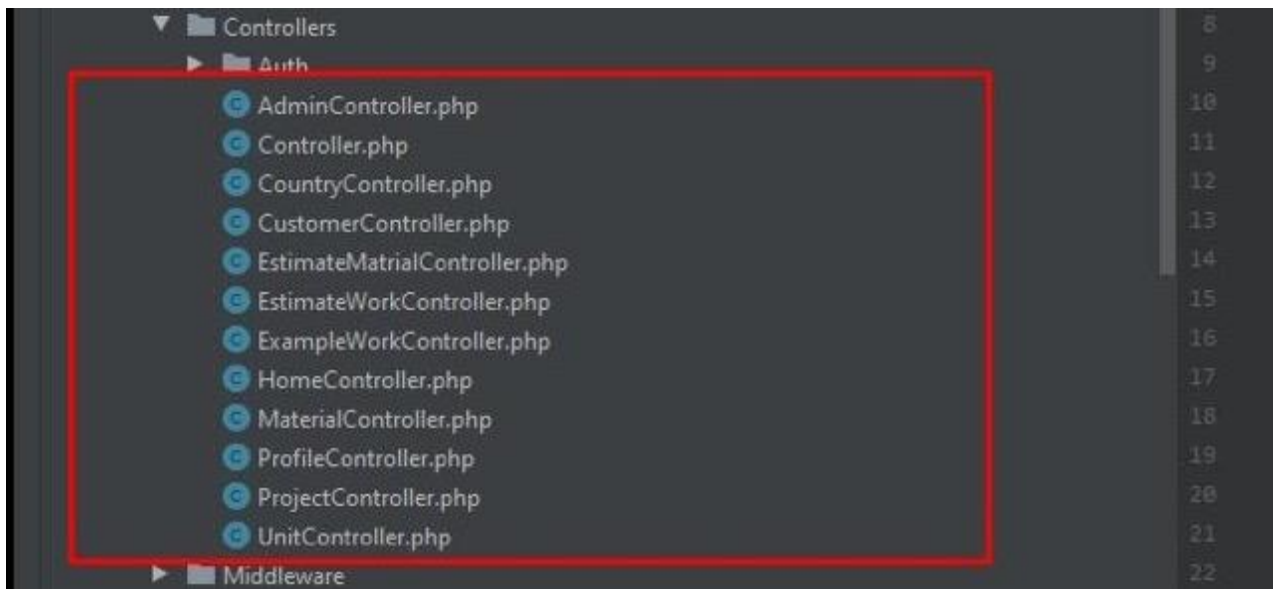


Рисунок 4.8 – Контролери

Після того, як відбулося підключення до бази даних, обрали всі необхідні дані, потім ці дані потрібно передати на сторінку. Для цього треба створити представлення views сторінок (рис. 4.9). Сторінки написані у поєднанні html та php коду, вони відображаються у браузері.

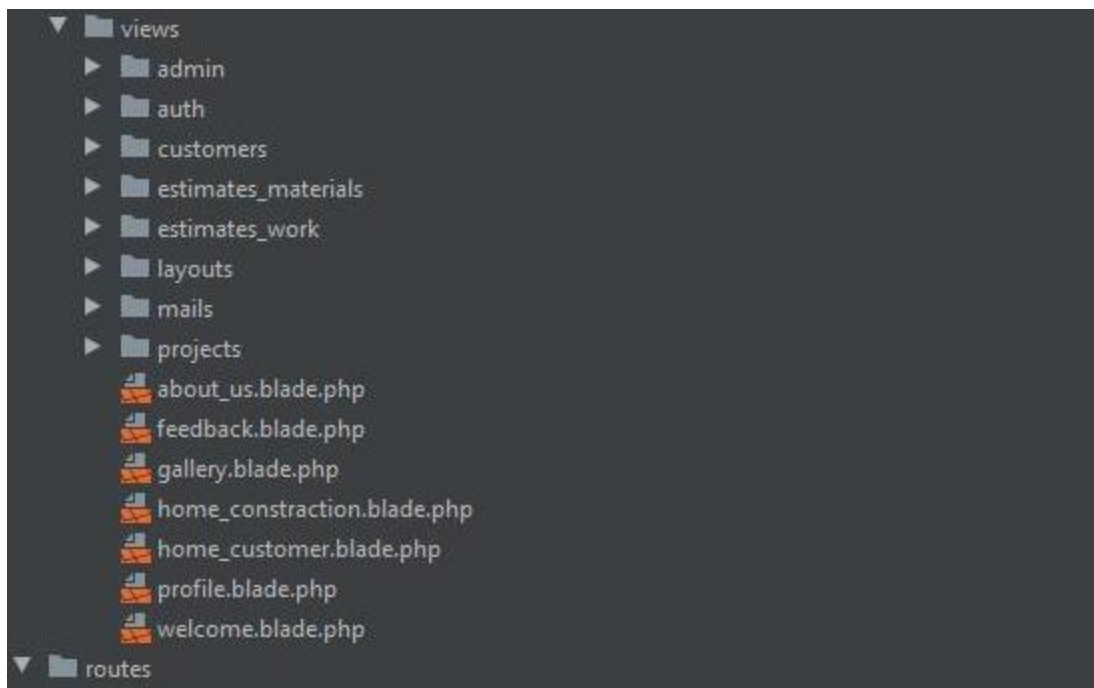


Рисунок 4.9 – Views

Крім views використовувався vue.js компонент, який відповідає за динаміку: натиснути на “+” – додається, натиснути на корзину – видаляється. (рис. 4.10).

Наприклад, компоненти estimateForm.vue, estimateMaterialForm.vue підключаються до views (в шаблон вбудовується компонент, передаються параметри).

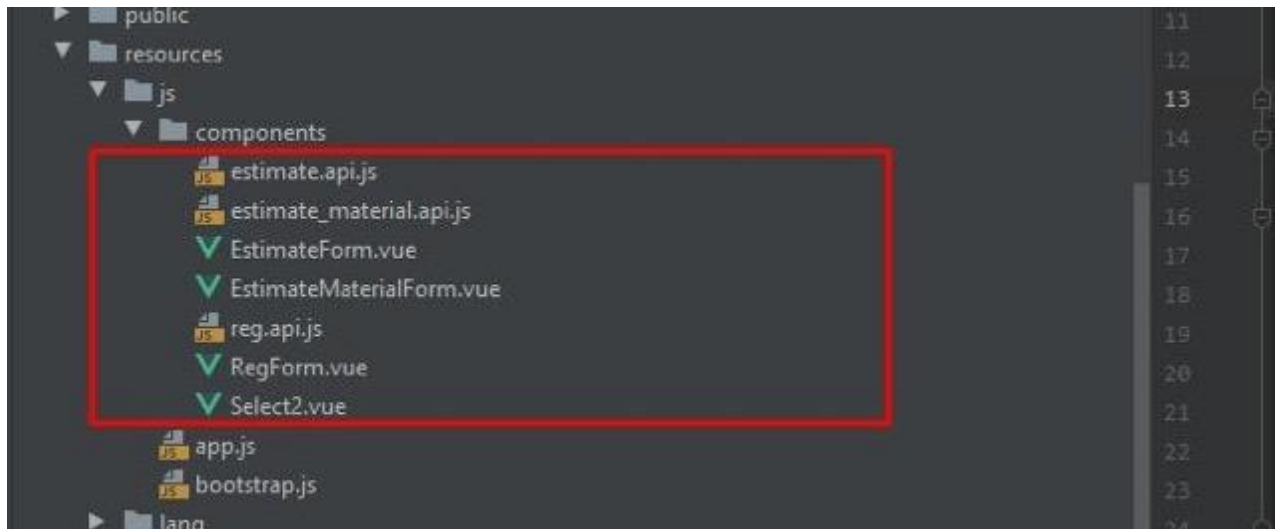


Рисунок 4.10 – Компоненти vue.js

Також важливим компонентом є routing. Routing – це система маршрутизації. Коли користувач, наприклад, тисне на кнопки, він здійснює маршрутизацію між сторінками. На рис. 4.11 показано, де саме налаштовується маршрутизація.

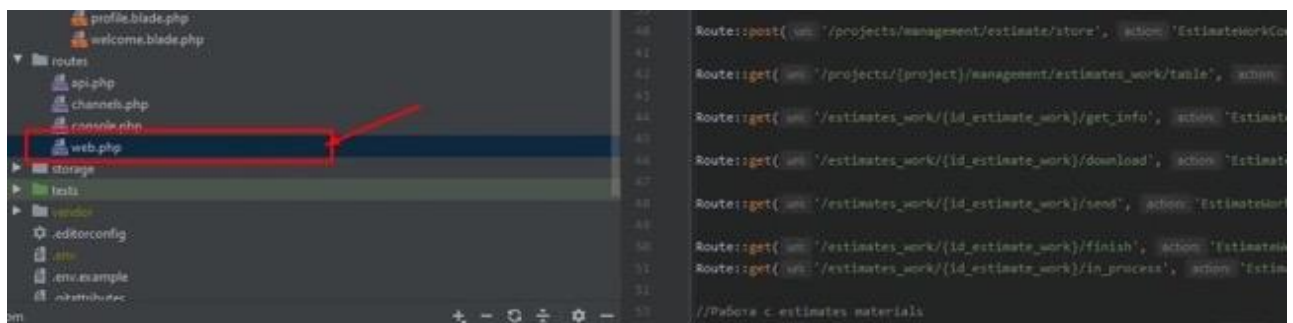


Рисунок 4.11 – Маршрутизація

Прописується назва роуту, його метод. Метод: get, post. Роути відповідальні, в який метод контролера він потрапляє.

Для того, щоб виконати команду маршрутизації використовується, наприклад, команда `php artisan router:cache` (скидається кеш та становляться видимими нові роути). Якщо додати роутер і не виконати цю команду, то нових роутів не буде видно (рис. 4.12).

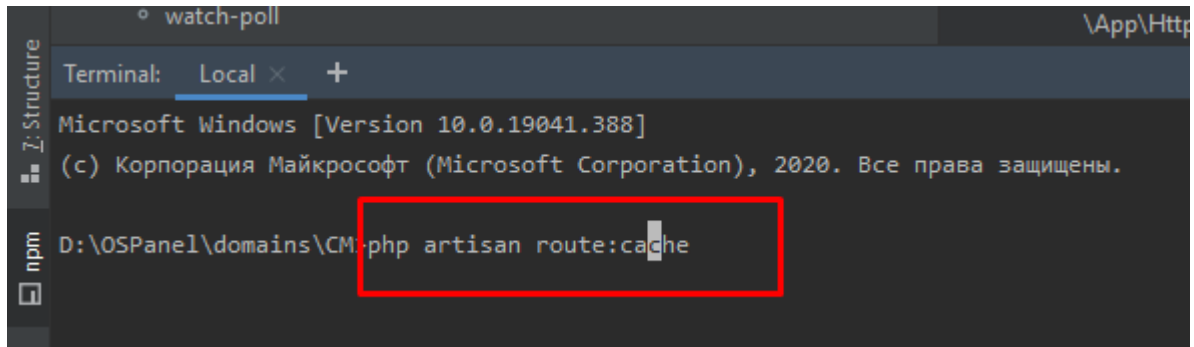
A screenshot of a terminal window in an IDE. The terminal title is 'watch-poll'. The window shows the command prompt 'D:\OSPanel\domains\CM>' followed by the command 'php artisan route:cache' which is highlighted with a red rectangular box. The terminal output shows the Windows version and copyright information for Microsoft Corporation, 2020.

Рисунок 4.12 – Команда очистки кешу

Laravel підтримує захист від кросбраузерних атак. Наприклад, коли відправляється форма, генерується тимчасовий token для унікальності даних, що відправляються (рис. 4.13), щоб неможливо було їх підробити. Без прописання token Laravel не дозволить нічого відправити на сервер і зберегти та буде видавати помилки. Для збереження оновлених даних token є обов'язковим.

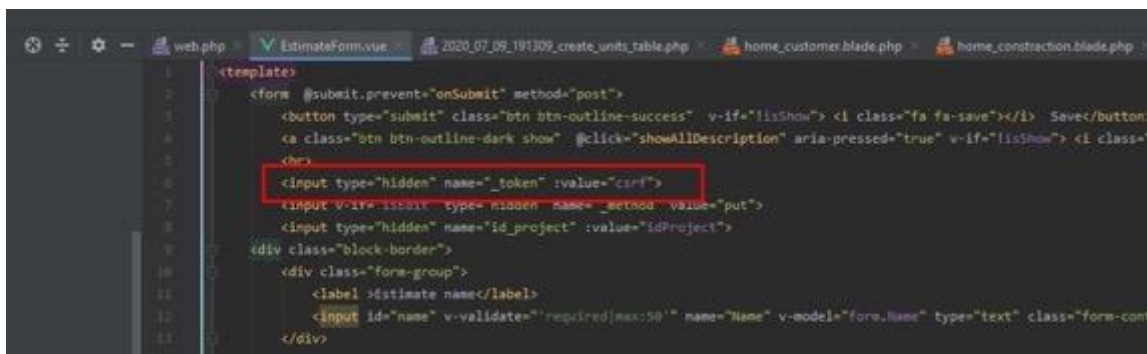
A screenshot of a code editor showing a Blade template snippet. The code is enclosed in a red rectangular box. The snippet includes a form with a submit button and several hidden input fields. One of the hidden input fields is highlighted with a red rectangular box, showing its name as '\_token' and its value as 'csrf'.

Рисунок 4.13 – Token



Laravel підтримує шифрування паролів (рис. 4.14).

The screenshot shows the phpMyAdmin interface with a table named 'users'. The table contains 11 rows of user data. The 'password' column shows that all passwords are hashed using bcrypt, starting with '\$2y\$10\$'. The table structure is as follows:

id	isAdmin	name	email	phone	email_verified_at	password
1	NUL	Martin Brown	al.grigorenko6@gmail.com	0958744001	NUL	\$2y\$10\$DQyAFzXUNCXK7H22baOSMG5age1CS2w4dy...
2	NUL	Kevin Harris	sashag2060@gmail.com	0956414852	NUL	\$2y\$10\$BwAMJ3M4y8yF0wP2.VYE1QzNR.p07x104C...
4	NUL	TTTT	sashag2561@gmail.com	4354334534	NUL	\$2y\$10\$BUPWjLorGTgl.DuX.n.y.P.koT0ma1%oa2aep...
5	1	Diga	diga.manchenko7@gmail.com	+380007820208	NUL	\$2y\$10\$9UVK6ajjVY5loR4YLarD3pQ8dN0jyT1FwY...
6	NUL	Tatiana	Taq.ta1@gmail.com	066447264	NUL	\$2y\$10\$4r1M8Ar6wWYyU8Kqf90Xpp8ndLZ3y0373...
7	NUL	Alena	lynska@yandex.ru	0994850511	NUL	\$2y\$10\$R1Uj.m8q2jzCGJY.SZ0079SF7Ym4z7pRkq...
8	NUL	TTTTTTTTTTTTTTTT	sasha@hanc.com	+380000202	NUL	\$2y\$10\$QpM3CAL0yUxH9j28oQ7QMPPhe8.LO...
9	NUL	32423	sas2122@gmail.com	304234234	NUL	\$2y\$10\$4j81TaDTEOR6JM7jwe.9k8S.eTKe8PKyY3M...
10	NUL	32423	sas24322@gmail.com	432423	NUL	\$2y\$10\$9w79613.eU8dLZymVVMANKES8NUG7MA7Z...
11	NUL	YUL7VA	julya.parkhenco@gmail.com	006090678	NUL	\$2y\$10\$RL7Dxl.x0qK0EFP5epQAR0kgRD1u3C0d...

Рисунок 4.14 – Шифрування паролів

Код формування кошторису сформований в додатку Б.

Для того, щоб отримати кошторис в pdf файлі, було підключено бібліотеку (рис. 4.15).

```

$ composer require barryvdh/laravel-dompdf
Warning: This development build of composer is over 60 days old. It is recommended to update
nel/modules/php/PHP_7.2-x64\composer.phar self-update" to get the latest version.
Using version ^0.8.6 for barryvdh/laravel-dompdf
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
 - Installing sabberworm/php-css-parser (8.3.1): Downloading (100%)
 - Installing phenx/php-svg-lib (v0.3.3): Downloading (100%)
 - Installing phenx/php-font-lib (0.5.2): Downloading (100%)
 - Installing dompdf/dompdf (v0.8.5): Downloading (100%)
 - Installing barryvdh/laravel-dompdf (v0.8.6): Downloading (100%)
dompdf/dompdf suggests installing ext-imagick (Improves image processing performance)
dompdf/dompdf suggests installing ext-gmagick (Improves image processing performance)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-dompdf
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.

```

Рисунок 4.15 – Підключення до бібліотеки для формування pdf файл

Для того, щоб рисунки мали однакову стилістику використовувався Photoshop (рис. 4.16).

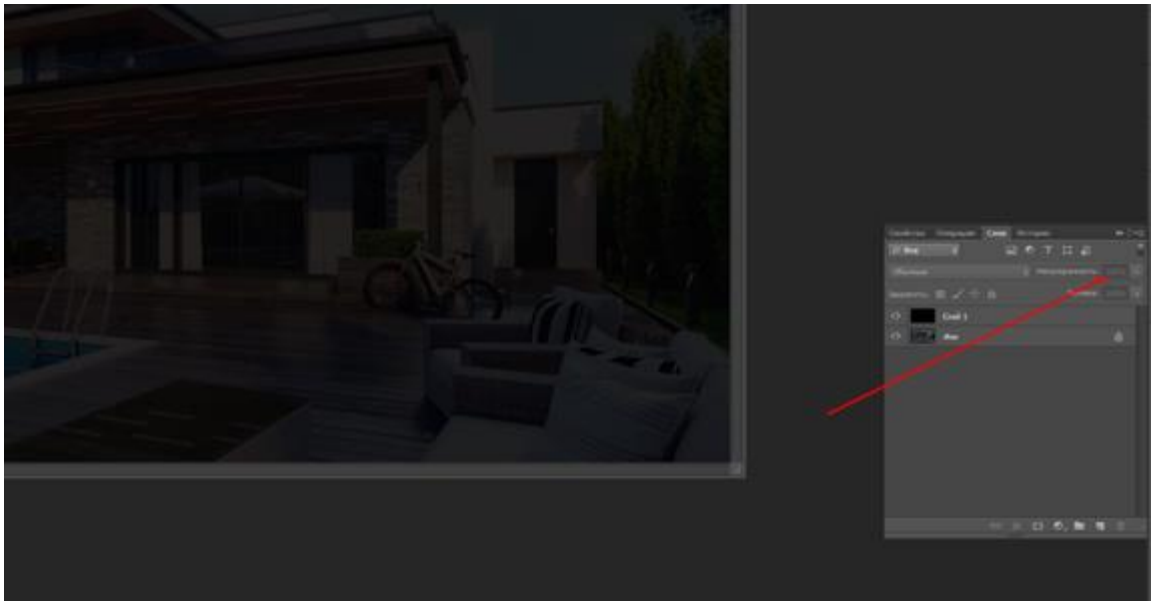


Рисунок 4.16 – Стилістика зображень

Після розробки проекту було проведено тестування, знайдені недоліки та виконанні виправлення (рис. 4.17).



Рисунок 4.17 – Виконання виправлень

### 4.3 Результат реалізації веб-орієнтованої інформаційної системи формування кошторису будівельних робіт

Для початку користувач вводить URL адресу сайту: <http://construction.ho.ua/>. Відкриється головна сторінка веб-сайту – Home (рис. 4.18). Дана сторінка є презентаційною, у верхній частині якої знаходиться меню: Home, About us, Gallery, Feedback, Login, Registration. Також вона містить в собі посилання на контактну інформацію та місцезнаходження через Google Map. Створенні посилання на соціальні мережі, такі як Messenger, Twitter, YouTube, Instagram та Facebook. Соціальні мережі на даний момент є не активованими, так як їх сторінки знаходяться в розробці у замовника. Важливо зазначити, що хостинг є безкоштовним, тому деякі функції неможливо буде відслідкувати.

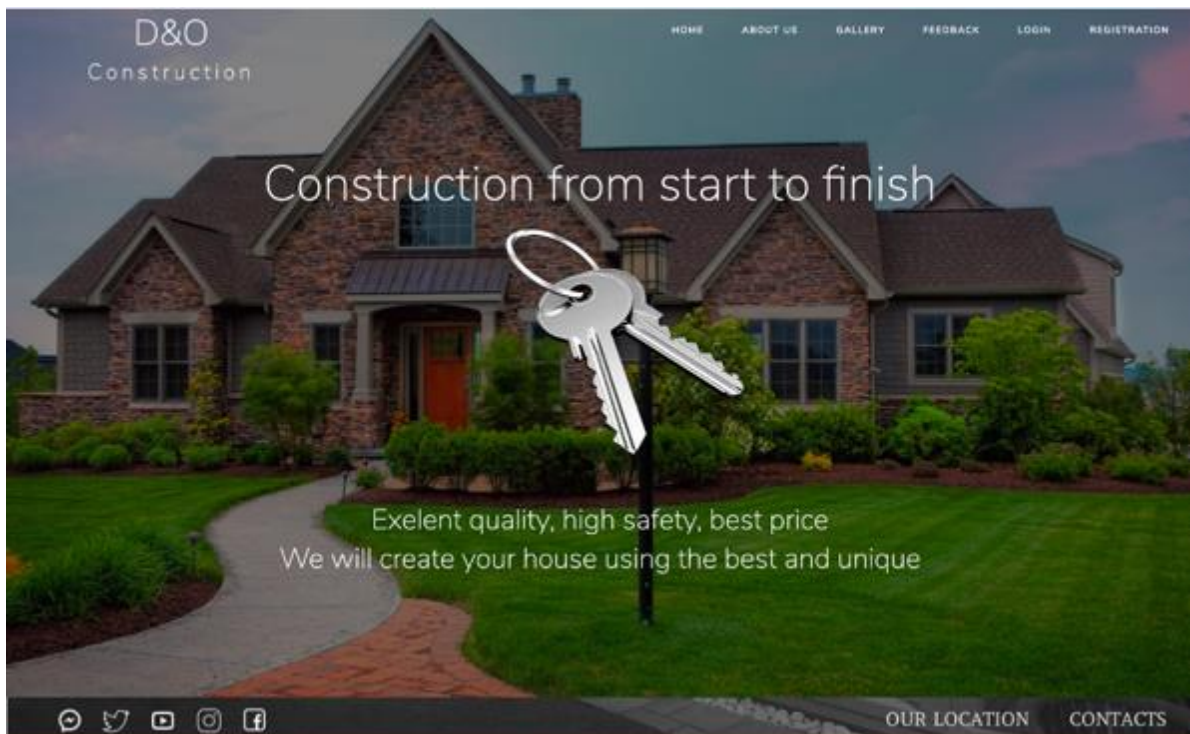


Рисунок 4.18 – Головна сторінка/HOME

Переходячи за посиланням About us, з'явиться сторінка про компанію (рис. 4.19).

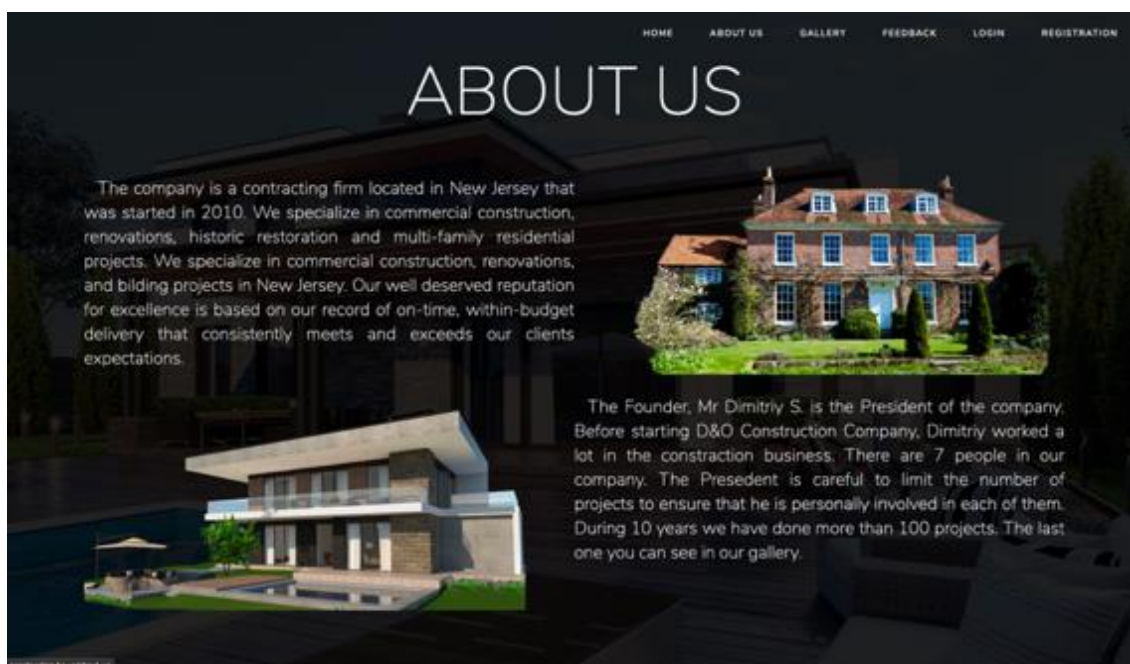


Рисунок 4.19 – Про компанію/About us

Наступне посилання відповідає за галерею (рис 4.20), де знаходяться фотозвіти робіт.

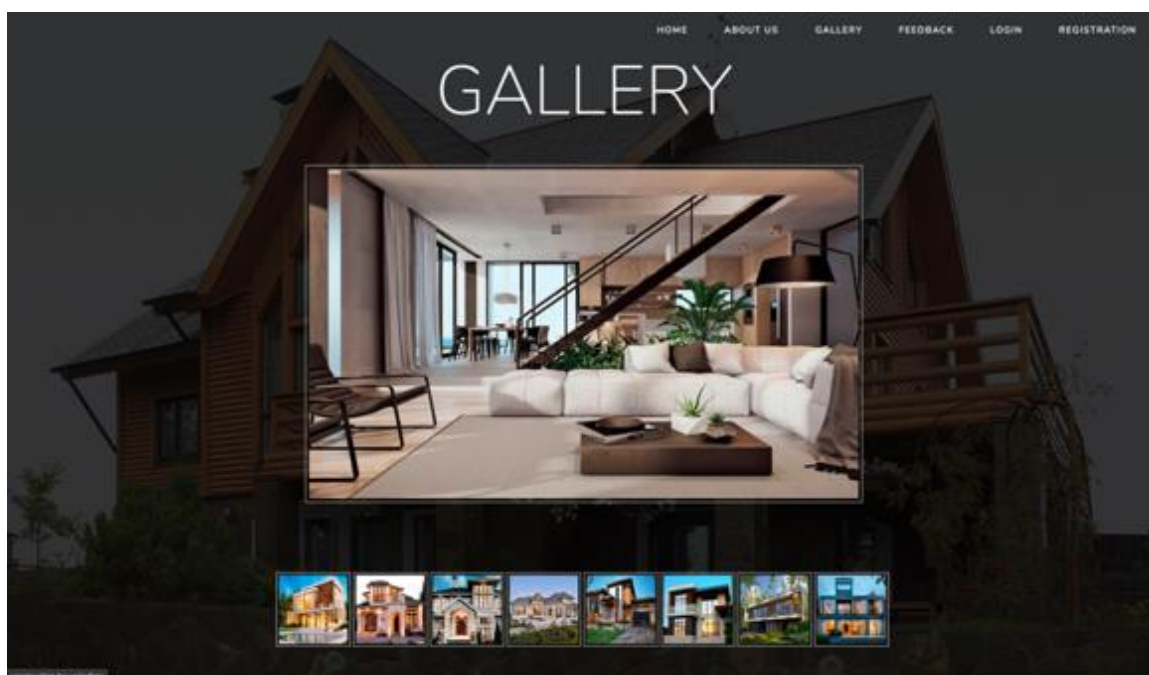


Рисунок 4.20 – Галерея/ Gallery

Посилання Feedback відповідає за перехід на сторінку відображення відгуків клієнтів, які вже скористалися послугами компанії (рис 4.21).

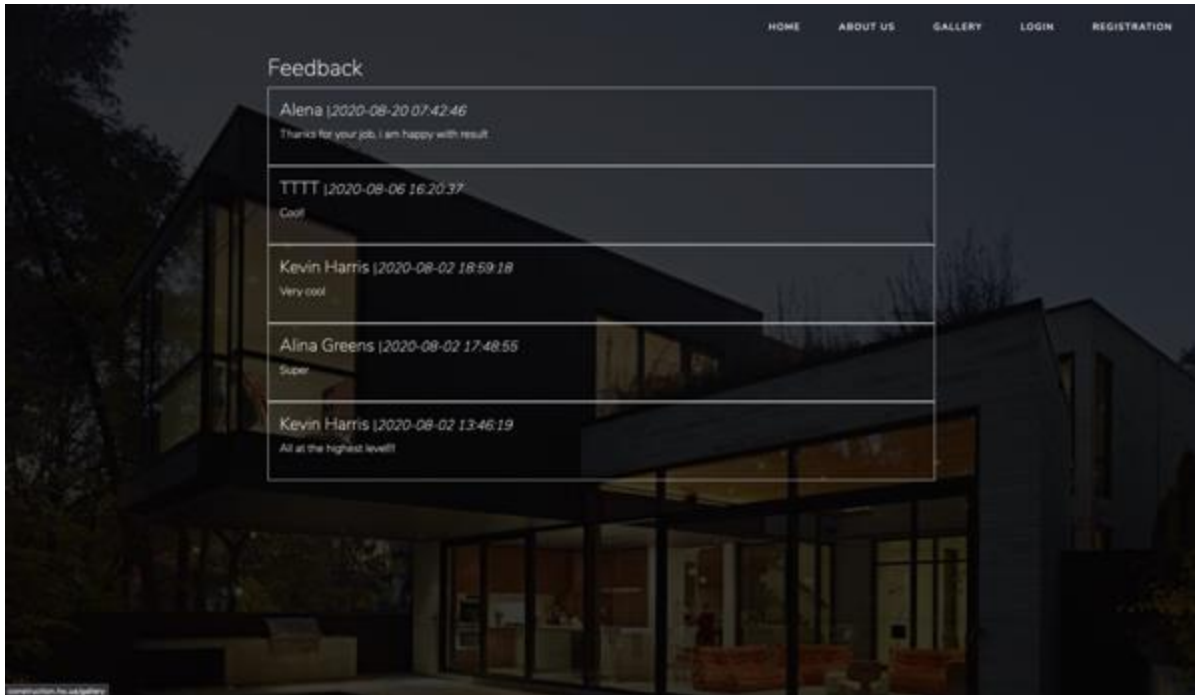


Рисунок 4.21 – Сторінка відгуків/ Feedback

Наступне посилання Our Location переходить на сторінку Google Map, відображаючи місцезнаходження компанії (рис. 4.22).

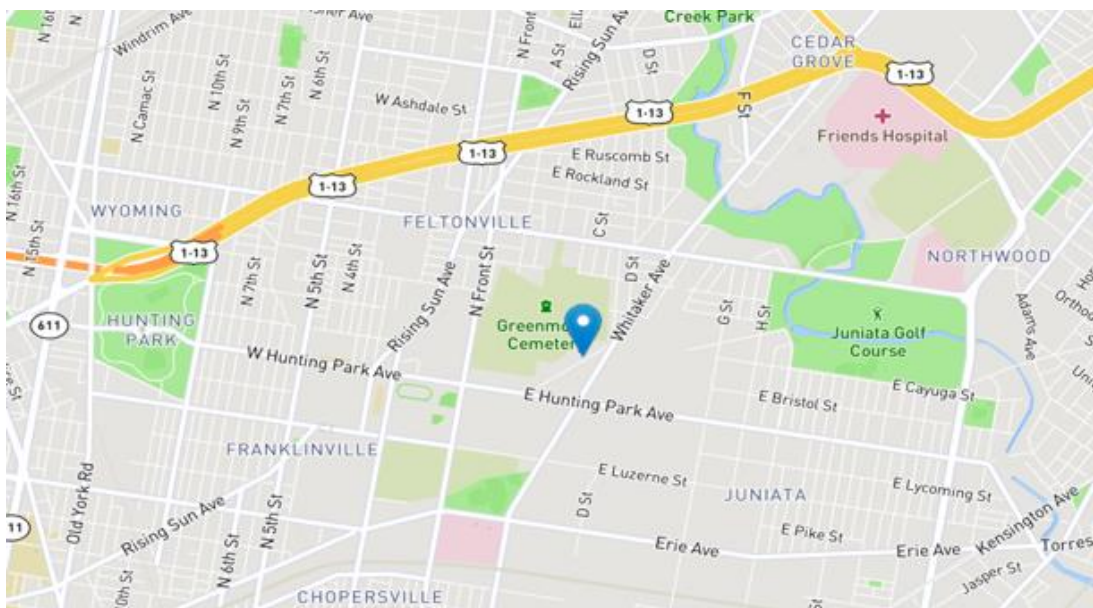


Рисунок 4.22 – Карта з місцезнаходженням компанії

Так як веб-сайт з авторизованим доступом та має розмежуванням на дві частини – клієнтську та адміністратора, реалізовані відповідні реєстрації. Переходячи за посиланням Registration, користувач обирає категорію до якої він відноситься: costumer або constructor (рис. 4.23).

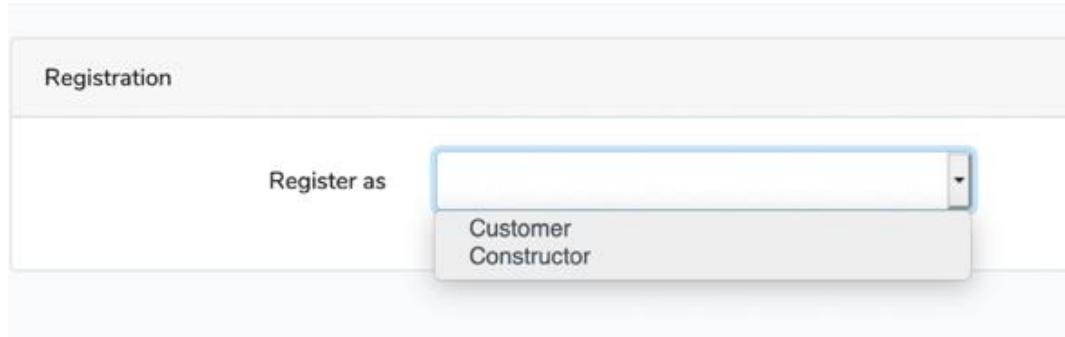
The image shows a web form titled "Registration". It contains a label "Register as" followed by a dropdown menu. The dropdown menu is open, showing two options: "Customer" and "Constructor".

Рисунок 4.23 – Реєстрація по категоріям

Заповнюються відповідні анкети (рис. 4.25, рис. 4.26). Анкети майже ідентичні, вони мають такі поля, як ім'я, електронна пошта, номер телефону, пароль (який має складатися з восьми символів), підтвердження пароля, адреса: країна, область/ регіон/ штат, місто, вулиця та індекс. Але є одна відмінність – це код, який складається з 4 цифр (рис. 4.24). Він створений, щоб обмежити небажаний доступ користувачів, які не мають відношення до представників виконавчої частини компанії. При правильному заповненні полів система подає зелений сигнал (рис. 4.26).

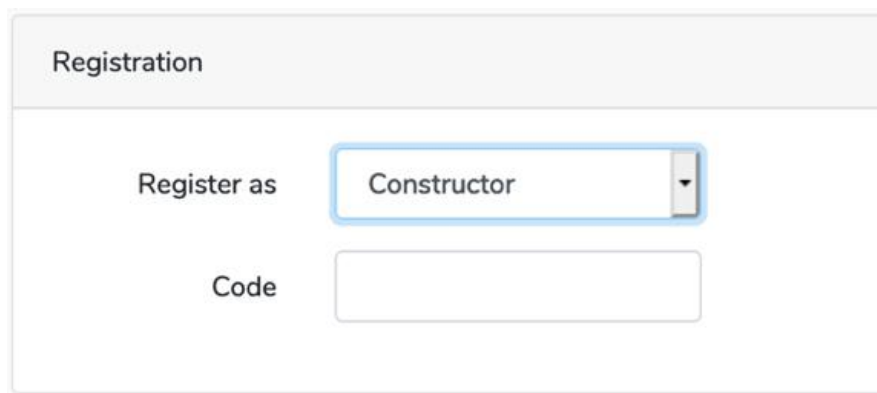
The image shows a web form titled "Registration". It contains a label "Register as" followed by a dropdown menu with "Constructor" selected. Below it is a label "Code" followed by an empty text input field.

Рисунок 4.24 – Обов'язковий код для реєстрації адміністратора

Registration

Register as

Name

Email

Phone

Password

Confirm Password

Address

Country

Region

City

Street

Zip

Рисунок 4.25– Анкета для клієнта/customer

Registration

Register as

Code

Name  ✓

Email  ✓

Phone

Password

Confirm Password

Address

Country

Region

City

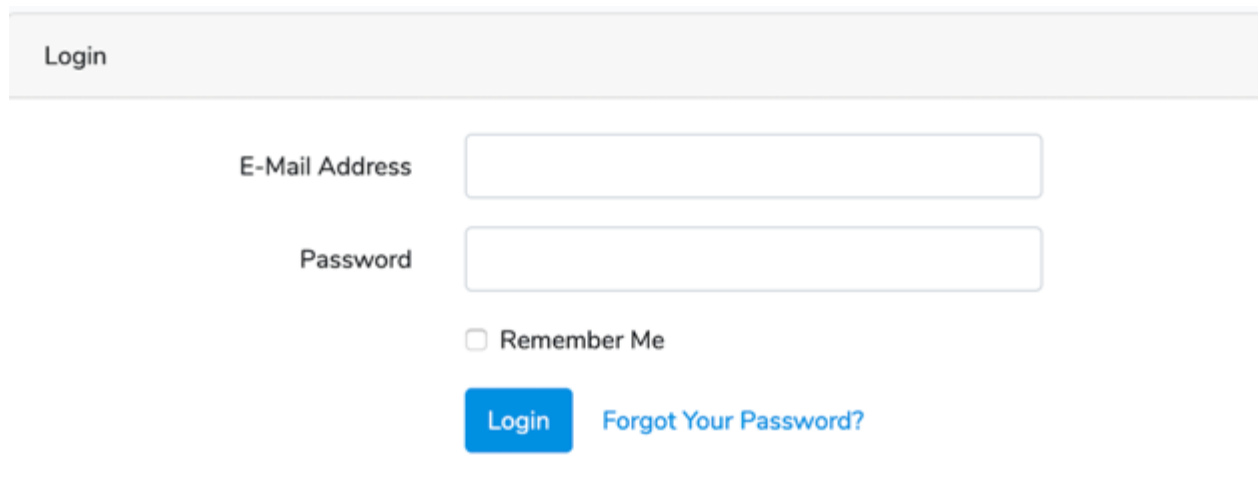
Street

Zip

Рисунок 4.26 – Анкета для виконавця/constructor

Після того, як користувач зареєструвався в системі, він має можливість зайти до свого аккаунту. Так як веб-сайт створений в першу чергу для формування кошторису будівельних робіт, розглянемо аккаунт адміністратора. Потрібно відмітити, що виконавець виконує функцію адміністратора.

Отже, щоб потрапити до кабінету, потрібно ввести електронну пошту та пароль (рис. 4.27). Для збереження своїх даних, щоб наступного разу входити автоматично, користувач може зберегти їх, використавши функцію Remember me (запам'ятати мене). У випадку, якщо пароль не запам'ятали можна звернутися до функції Fogot Your Password. При цьому потрібно буде ввести свою електронну адресу, на яку система відправить посилання, через яке можна буде змінити пароль.



The image shows a login form with a light gray header containing the word "Login". Below the header, there are two input fields: "E-Mail Address" and "Password". Below the "Password" field, there is a checkbox labeled "Remember Me". At the bottom of the form, there is a blue button labeled "Login" and a link labeled "Forgot Your Password?".

Рисунок 4.27 – Вхід до кабінету

Потрапивши в кабінет адміністратора, з'явиться робоча панель, яка містить в собі наступне меню: Projects, My Customers, Admitistration (рис. 4.28). У правому верхньому кутку знаходяться Profile та Logout. Заходячи в Profile, можна редагувати налаштування, які відповідають за особисту інформацію адміністратора та за Secret Code, який надається при реєстрації представникам компанії (рис. 4.29).



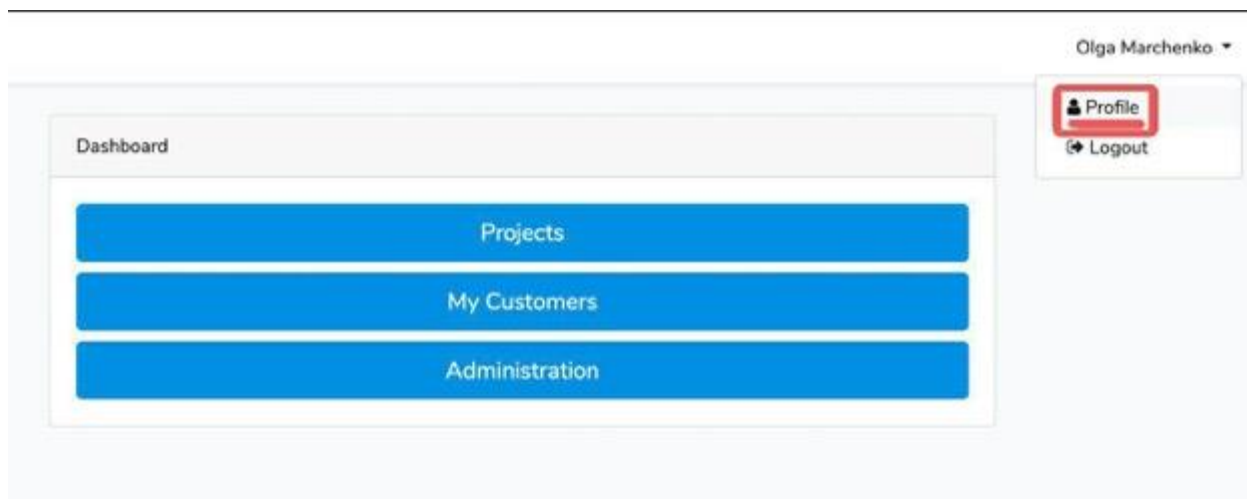


Рисунок 4.28 – Робоча панель акаунту адміністратора

The screenshot displays the "Profile settings" page. At the top right, the user's name "Olga Marchenko" is shown. A "Back" button is located in the top right corner of the settings area. The form contains the following fields:

- Name: Olga Marchenko
- E-Mail Address: [Redacted]
- Phone: [Redacted]
- Country: Ukraine (dropdown menu)
- Region: Sumy
- City: Sumy
- Street: [Redacted]
- Zip code: 40009
- Secret Code: [Redacted]

A "Save" button is positioned at the bottom of the form.

Рисунок 4.29 – Налаштування особистих даних адміністратора

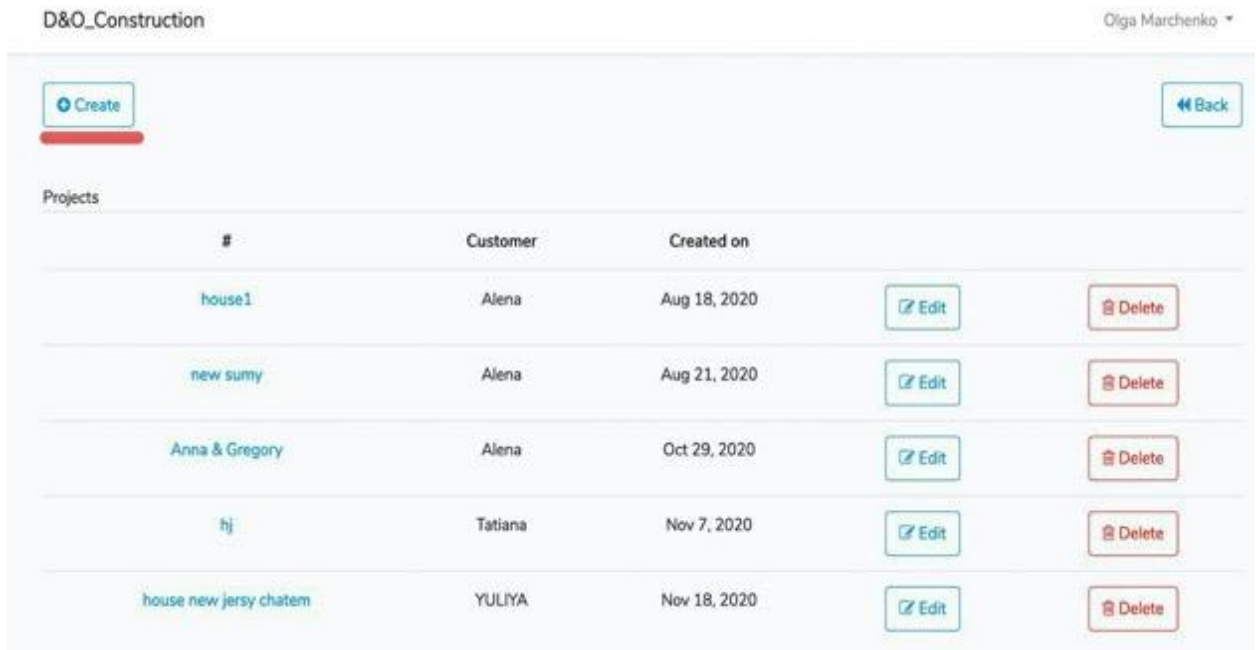


Рисунок 4.30 – Projects

Переходячи в меню Projects, система пропонує створити новий проект або перейти до вже створених (рис. 4.30). Проекти можна редагувати (назву та замовника) чи видаляти. Для створення нового проекту потрібно натиснути на Create. З'явиться нова сторінка, на якій адміністратор вказує назву нового проекту та обирає клієнта, який вже занесений в базу даних (рис. 4.31).

D&O\_Construction

New project

Customer

Kevin Harris

Create project

Рисунок 4.31 – Створення нового проекту

Відкривши один із проектів, система виконує перехід на сторінку Project management/управління проектом (рис. 4.32). Саме тут адміністратор має можливість створити кошториси робіт та кошториси на матеріали: Make Estimate of Work, Make Estimate of Materials; та переглянути їх в блоках мої кошториси робіт та мої кошториси матеріалів (My Estimates of Work, My Estimates of Materials).

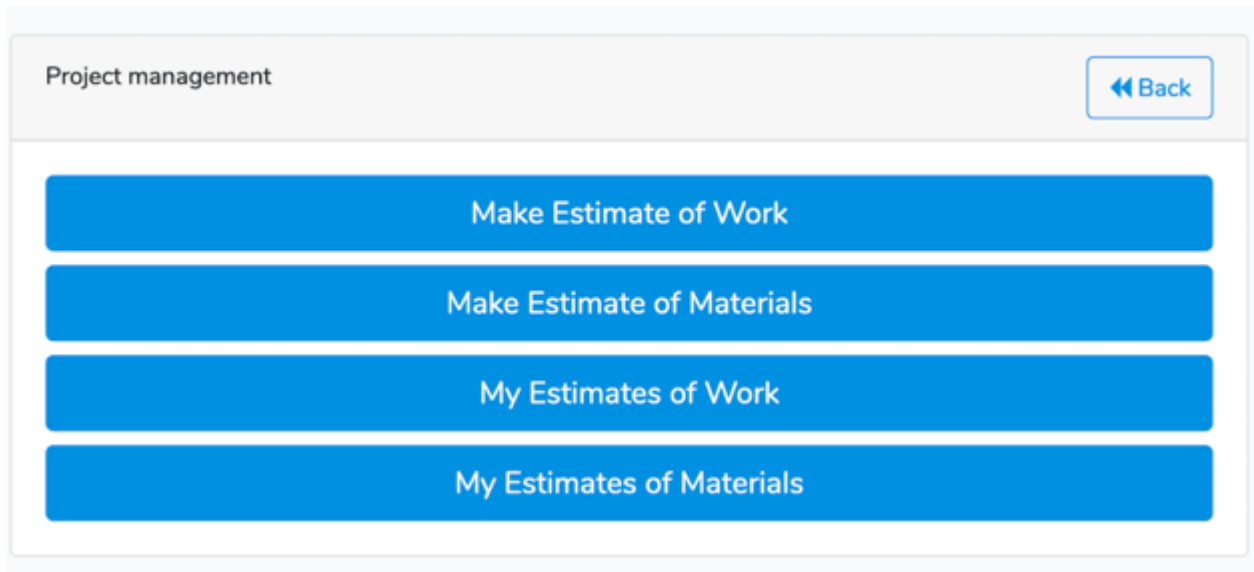


Рисунок 4.32 – Project management/управління проектом

The screenshot shows the "Make Estimate of Work" form. At the top left, the title "Make Estimate of Work" is displayed. In the top right corner, there is a blue button with a left-pointing arrow and the text "Back". Below the title bar, there are two buttons: a green "Save" button and a "Show all description" button. Underneath, there is an "Estimate name" label followed by a text input field. Below that, there are two input fields: "Max price" and "Total price" (with a value of "0"). The main part of the form is a table with the following columns: "#", "Work type", "Description", "Unit", "Price", "Quantity", and "Total amount". The first row of the table has the number "1" in the "#" column, and dropdown menus in the "Work type", "Description", and "Unit" columns. The "Price" column has a text input field with a "\$" symbol. The "Quantity" and "Total amount" columns have text input fields. At the bottom of the form, there is a large blue button with a plus sign and the text "Add work".

Рисунок 4.33 – Make Estimate of Work

Переходячи в Make Estimate of Work (рис. 4.33), отримуємо форму для заповнення кошторису. У формі потрібно заповнити наступні поля: найменування кошторису, Max price (максимальну ціна, на яку розраховує замовник), Total price (загальна сума коштів за роботу). Таблиця підрахунку включає в себе: Work type (види робіт), які поділяються на interior (внутрішні), exterior (зовнішні) і Furniture Assembly (робота з фурнітурою); Description (опис роботи); Unit (одиниці вимірювання); Price (ціна за 1 одиницю вимірювання); Quantity (кількість); Total amount (всього по даному типу роботи). Одиниці вимірювання та ціну за одну одиницю вимірювання можна отримати автоматично, натиснувши відповідні кнопки в цих колонках, або внести самостійно. “+Add work” додає в таблицю нове поле для заповнення. Всі поля повинні бути заповнені, інакше, система повідомить про помилку.

Розглянемо процес створення кошторису (рис. 4.34).

Make Estimate of Work ← Back

Save Show all description

Estimate name: 1 ✓

Max price: 1500 Total price ▲ 1578

#	Work type	Description	Unit	Price	Quantity	Total amount
1	Exterior		sf	6 \$	7 ✓	42.00
2	Exterior		sf	10 \$	6 ✓	60.00
3	Exterior		yd	10 \$	9 ✓	90.00
4	Exterior		yd	150 \$	5 ✓	750.00
5	Interior		sf	1 \$	7 ✓	7.00
6	Interior		sf	500 \$	1 ✓	500.00
7	Furniture A:	Install cabinet's r	ea	3 \$	3 ✓	9.00
8	Furniture A:	Install drawer	ps	30 \$	4 ✓	120.00

+ Add work

Рисунок 4.34 – Формування кошторису

У стовпчику Description на рис. 4.34 існують поля, які не відображають інформацію. В цьому випадку потрібно натиснути кнопку Show all description, яка знаходиться над формою. Після цієї дії всі поля будуть заповнені (рис. 4.35).

Make Estimate of Work ← Back

Save Show all description

Estimate name: 1 ✓

Max price: 1500 Total price ⚠ 1578

#	Work type	Description	Unit	Price	Quantity	Total amount
1	Exterior	Install decking	sf	6 \$	7 ✓	42.00
2	Exterior	Roofing asphalt	sf	10 \$	6 ✓	60.00
3	Exterior	Excavation	yd	10 \$	9 ✓	90.00
4	Exterior	Garbage removal	yd	150 \$	5 ✓	750.00
5	Interior	Prime walls and ceiling	sf	1 \$	7 ✓	7.00
6	Interior	Install new shower	sf	500 \$	1 ✓	500.00
7	Furniture A	Install cabinet's n	ea	3 \$	3 ✓	9.00
8	Furniture A	Install drawer	ps	30 \$	4 ✓	120.00

+ Add work

Рисунок 4.35 – Результат заповнення усіх полів

Якщо загальна сума кошторису перевищує Max price, система повідомить про помилку. Щоб видалити рядок, потрібно натиснути на значок корзини, яка знаходиться в першому стовпчику таблиці.

Коли всі помилки виправлені, всі питання щодо ремонтно-будівельних робіт узгодженні, кошторис зберігається, натиснувши кнопку Save. Документ в форматі PDF (рис. 4.36) відразу відображається в системі та зберігається в My Estimate of Work.

**Constructor**

Olga Marchenko  
 Ukraine, Sumy, Sumy, 40009  
 [redacted]@gmail.com  
 +38 [redacted]

**Customer**

YULIYA  
 Ukraine, sumy, sumy, 40000  
 [redacted]@gmail.com  
 050 [redacted]

**Estimate of work**

#	Description of work	Quantity	Total amount
1	Install deking	7 sf	42 \$
2	Roofing asphalt shingles	6 sf	60 \$
3	Excavation	9 yd	90 \$
4	Garbage removal	5 yd	750 \$
5	Prime walls and ceiling	7 sf	7 \$
6	Install new shower	1 sf	500 \$
7	Install cabinet's molding	3 ea	9 \$
		<b>Total</b>	<b>1458 \$</b>

Customer \_\_\_\_\_

Date: 11. 18. 2020

Constructor \_\_\_\_\_

Рисунок 4.36 – Кошторис робіт

Розглянемо меню Make Estimate of Materials (рис. 4.37). Форма заповнення схожа з попередньою. Обов'язково заповнюються всі поля. Таблиця складається з Description (опис матеріалу), Unit, Price, Quantity, Total amount. Ціна є фіксованою, але у випадку об'ємних замовлень її можна корегувати самостійно.

Make Estimate of Materials ← Back

Save

Estimate name

Max price  Total price 0

#	Description	Unit	Price	Quantity	Total amount
1	<input type="text"/>	<input type="text"/>	<input type="text"/> \$	<input type="text"/>	<input type="text"/>

+ Add work

Рисунок 4.37 – Make Estimate of Materials

Кошторис неможливо буде зберегти, якщо система відображає помилки (рис. 4.38).

Make Estimate of Materials ← Back

Save

Estimate name

1b ✓

Max price 500 Total price ▲ 790

#	Description	Unit	Price	Quantity	Total amount
1	Brick white	sf	12 \$	20 ✓	240.00
2	Cement 500	ps	3.5 \$	10 ✓	35.00
3	Door white 36**80"	ea	50 \$	2 ✓	100.00
4	Facade white	sf	5 \$	5 ✓	25.00
5	Drywall	sf	13 \$	30 ✓	390.00

+ Add work

Рисунок 4.38 – Складання кошторису по матеріалам, відображення помилок

При збереженні інформації, як результат, отримаємо файл в форматі PDF (рис. 4.39). Він буде знаходитися в My Estimate of Materials.

<p><b>Constructor</b></p> <p>Olga Marchenko Ukraine, Sumy, Sumy, 40009 +380 [redacted] [redacted]@gmail.com</p>	<p><b>Customer</b></p> <p>YULIYA Ukraine, sumy, sumy, 40000 [redacted] [redacted]@gmail.com 05 [redacted]</p>
---	---

### Estimate of Materials

#	Description of material	Quantity	Total amount
1	Brick white	20 sf	240 \$
2	Cement 500	10 ps	35 \$
3	Door white 36"*80"	2 ea	100 \$
4	Facade white	5 sf	25 \$
		<b>Total</b>	<b>400 \$</b>

Customer \_\_\_\_\_

Date: 11. 18. 2020

Constructor \_\_\_\_\_

Рисунок 4.39 – Кошторис по матеріалам

Розглянемо меню My Estimates of Work. У ньому відображений список усіх кошторисів будівельних робіт (первинні, узгоджені, екстра кошториси) відповідного проекту (рис. 4.40). Список складається з наступної інформації: назва кошторису, дата створення, статус (New, In process, Done), коригування кошторису та його видання.

Натиснувши на ім'я кошторису, відображається сторінка з повною інформацією про нього. Документ можна завантажити (Download PDF) або відправити на електронну пошту замовнику, натиснувши кнопку Send PDF



(рис. 4.41). Електронна адреса підтягнеться автоматично. На рис. 4.42 зображено отримання електронної пошти користувачем з кошторисом у вигляді файлу PDF.

Новостворений кошторис має статус New. Якщо замовник його погоджує, то він змінює статус на In process, натиснувши на зелену галочку, яка знаходиться поряд справа. При умові, що на проекті завершені ремонтно-будівельні роботи, адміністратор встановлює статус Done.

Натиснувши на кнопку Edit, система переведе на сторінку, яка відповідає за корегування кошторису (рис. 4.43). Натиснувши кнопку Delete, система видалить кошторис.

Estimates of Work					
#	Created on	Status			
1 basic	Nov 18, 2020	New	<input checked="" type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Delete</a>
1	Nov 18, 2020	Done	<input checked="" type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Delete</a>

Рисунок 4.40 – My Estimates of Work

Show Estimate of Work																											
#	Work type	Description	Unit	Price	Quantity	Total amount																					
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>Estimate name</span> <span>1 basic <input checked="" type="checkbox"/></span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span>Max price</span> <span>Total price 1948</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Work type</th> <th>Description</th> <th>Unit</th> <th>Price</th> <th>Quantity</th> <th>Total amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Exterior</td> <td>Framing</td> <td>sf</td> <td>2</td> <td>4</td> <td>8</td> </tr> <tr> <td>2</td> <td>Exterior</td> <td>Pavers</td> <td>sf</td> <td>11</td> <td>10</td> <td>110</td> </tr> </tbody> </table>							#	Work type	Description	Unit	Price	Quantity	Total amount	1	Exterior	Framing	sf	2	4	8	2	Exterior	Pavers	sf	11	10	110
#	Work type	Description	Unit	Price	Quantity	Total amount																					
1	Exterior	Framing	sf	2	4	8																					
2	Exterior	Pavers	sf	11	10	110																					

Рисунок 4.41 – Функції відправки PDF файлу на електронну пошту та його завантаження

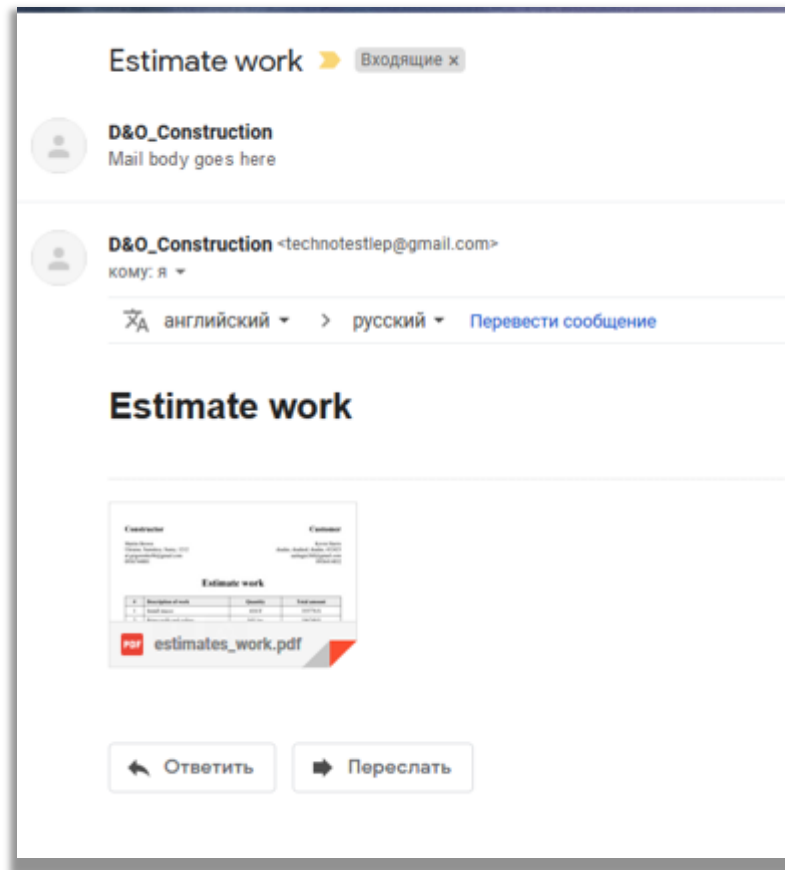


Рисунок 4.42 – Отримання кошторису на електронну пошту

[Save](#)
[Show all description](#)

Estimate name

1
✓

Max price

Total price 1458

#	Work type	Description	Unit	Price	Quantity	Total amount
1	Exterior	Install decking	sf	6 \$	7	42
2	Exterior	Roofing asphalt	sf	10 \$	6	60
3	Exterior	Excavation	yd	10 \$	9	90
4	Exterior	Garbage removal	yd	150 \$	5	750
5	Interior	Prime walls and ceiling	sf	1 \$	7	7
6	Interior	Install new shower	sf	500 \$	1	500
7	Furniture A	Install cabinet's	ea	3 \$	3	9

[+ Add work](#)

Рисунок 4.43 – Коригування кошторису робіт

Розглянемо меню My Estimates of Materials (рис. 4.44). Аналогічна схема відображення списку, зміни статусу, коригування (рис. 4.45) та видалення кошторисів.

Estimates of Materials

#	Created on	Status			
mater 1	Nov 18, 2020	Done	✓	<a href="#">Edit</a>	<a href="#">Delete</a>
mater 1	Nov 18, 2020	New	✓	<a href="#">Edit</a>	<a href="#">Delete</a>
1b	Nov 18, 2020	New	✓	<a href="#">Edit</a>	<a href="#">Delete</a>

Рисунок 4.44 – My Estimates of Materials

Edit Estimate of Materials [← Back](#)

[Save](#)

Estimate name

1b ✓

Max price  Total price 400

#	Description	Unit	Price	Quantity	Total amount
1	Brick white	sf	12 \$	20	240
2	Cement 500	ps	3.5 \$	10	35
3	Door white 36**80"	ea	50 \$	2	100
4	Facade white	sf	5 \$	5	25

[+ Add work](#)

Рисунок 4.45 – Коригування кошторису матеріалів

Повернемося до головної панелі акаунту та перейдемо в меню My Customers (рис. 4.46). На новій сторінці відобразиться список авторизованих клієнтів. Відкривши одного із клієнтів, побачимо закріплені за ним проекти (назва проекту, дата створення) (рис. 4.47). У кожному із проектів сформовані

списки кошторисів за двома типами: кошториси будівельних робіт та кошториси на матеріали (рис. 4.48). У списку відображається інформація: найменування проекту (Title), дата створення (created on) і статус (Status).

My Customers
<a href="#">Alena</a>
<a href="#">Tatiana</a>
<a href="#">YULIYA</a>

Рисунок 4.46 – Список клієнтів

Projects	#	Created on
<a href="#">house new jersy chatem</a>		Nov 18, 2020

Рисунок 4.47 – Список проектів, які закріплені за клієнтом

Estimates of Work				
Title	Created on	Status		
<a href="#">1 basic</a>	Nov 18, 2020	New	<input checked="" type="checkbox"/>	
<a href="#">1</a>	Nov 18, 2020	Done	<input checked="" type="checkbox"/>	
Estimates of Materials				
Title	Created on	Status		
<a href="#">mater 1</a>	Nov 18, 2020	Done	<input checked="" type="checkbox"/>	
<a href="#">mater 1</a>	Nov 18, 2020	New	<input checked="" type="checkbox"/>	
<a href="#">1b</a>	Nov 18, 2020	New	<input checked="" type="checkbox"/>	

Рисунок 4.48 – Список кошторисів по проекту

Натиснувши на назву кошторису, відкриється його файл у форматі PDF.

Розглянемо адміністративну частину (Administration) кабінету (рис. 4.49). Адміністративне меню відповідає за внесення змін до бази даних. Воно складається з Work examples (прикладі робіт або описання робіт), Materials (матеріали), Units (одиниці вимірювання) та Countris (країни).

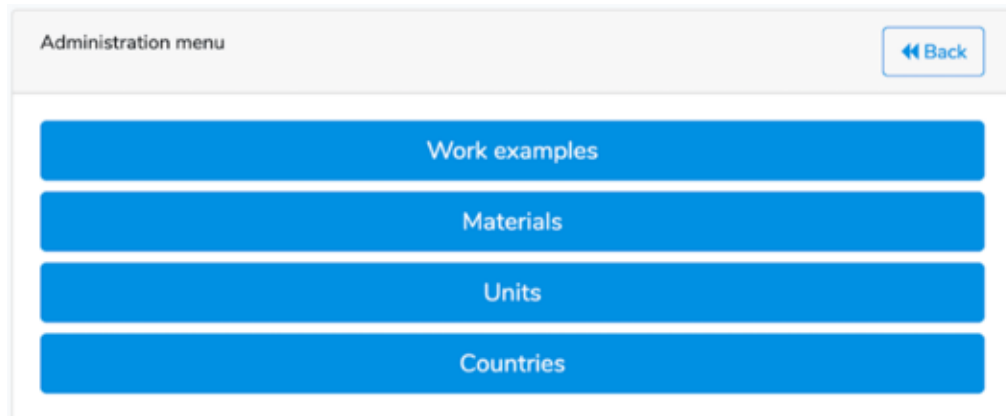


Рисунок 4.49 – Адміністративне меню

Сторінка Work examples містить інформацію про роботу, яку може здійснювати виконавець (рис. 4.50). Адміністратор має можливість переглянути список робіт, за необхідності використовується функція пошуку або фільтри алфавітного значення (від А до Я, від Я до А); відкорегувати інформацію, натиснувши на кнопку Edit (рис. 4.51) або видалити її, натиснувши кнопку Delete; переглянути ціни, використовуючи фільтр пошукового діапазону або фільтри від меншої ціни до більшої, від більшої до меншої; та доповнити основний список, натиснувши на значок + (рис. 4.52) і заповнивши форму. В неї можна винести опис роботи, ціну, одиниці вимірювання і тип роботи. Збереження інформації відбувається, натиснувши кнопку Save.

Work examples

Search

Description	Price	Type work	Unit base		
Garbage removal	150 \$	Exterior	yd	<a href="#">Edit</a>	<a href="#">Delete</a>
Install stucco	10 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>
Excavation	10 \$	Exterior	yd	<a href="#">Edit</a>	<a href="#">Delete</a>
Concrete pouring	65 \$	Exterior	yd	<a href="#">Edit</a>	<a href="#">Delete</a>
Framing	3 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>
Roofing asphalt shingles	10 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>
Install decking	6 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>
Install railing	3 \$	Exterior	lf	<a href="#">Edit</a>	<a href="#">Delete</a>
Pavers	12 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>
Power wash	3 \$	Exterior	sf	<a href="#">Edit</a>	<a href="#">Delete</a>

Рисунок 4.50 – Сторінка з інформацією про види робіт

Description

Price

Type work

Exterior

Unit base

sf

Рисунок 4.51 – Додавання опису робіт

Description

Garbage removal

Price

150

Type work

- ✓ Exterior
- Interior
- Furniture Assembly

Unit base

yd

Save

Рисунок 4.52 – Корегування опису робіт

Сторінка Materials містить інформацію про матеріали (рис. 4.53). Адміністратор має можливість переглянути базу даних по матеріалам, їх цінову політику, використовуючи пошук, фільтр пошуку, фільтр цін, поповнювати базу, додаючи інформацію через кнопку “+”, видаляти матеріали та коригувати їх (рис. 4.54).

Materials

Search

+ [List] [Filter] [Filter] [Filter]

from to

Description	Price		
Brick red	10 \$	Edit	Delete
Brick white	12 \$	Edit	Delete
Cement 400	3 \$	Edit	Delete
Cement 500	3.5 \$	Edit	Delete

Рисунок 4.53 – Materials



Description

Brick red

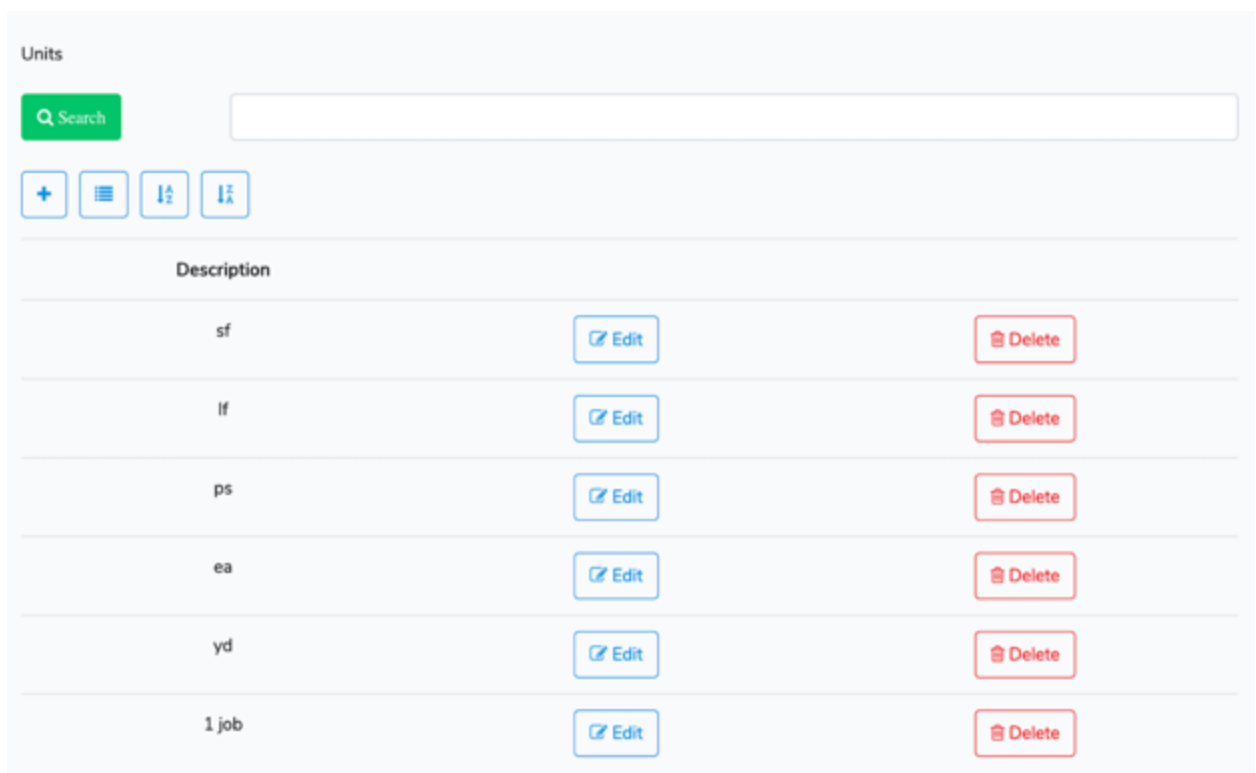
Price

10

Save

Рисунок 4.54 – Корегування матеріалів

Сторінка Units (рис. 4.55) – база даних одиниць вимірювань. Адміністратор може додавати, корегувати (рис. 4.56) та видаляти інформацію.



Units

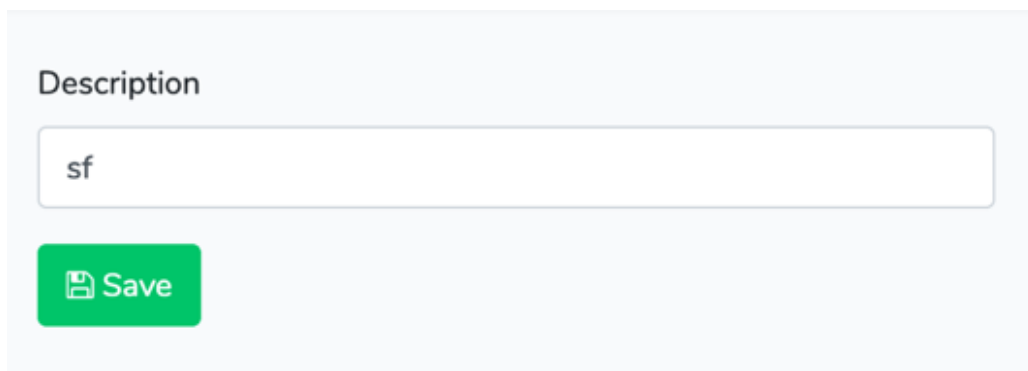
Search

+ [List Icon] [Filter Icon] [Sort Icon]

Description		
sf	Edit	Delete
lf	Edit	Delete
ps	Edit	Delete
ea	Edit	Delete
yd	Edit	Delete
1 job	Edit	Delete

Рисунок 4.55 – Units



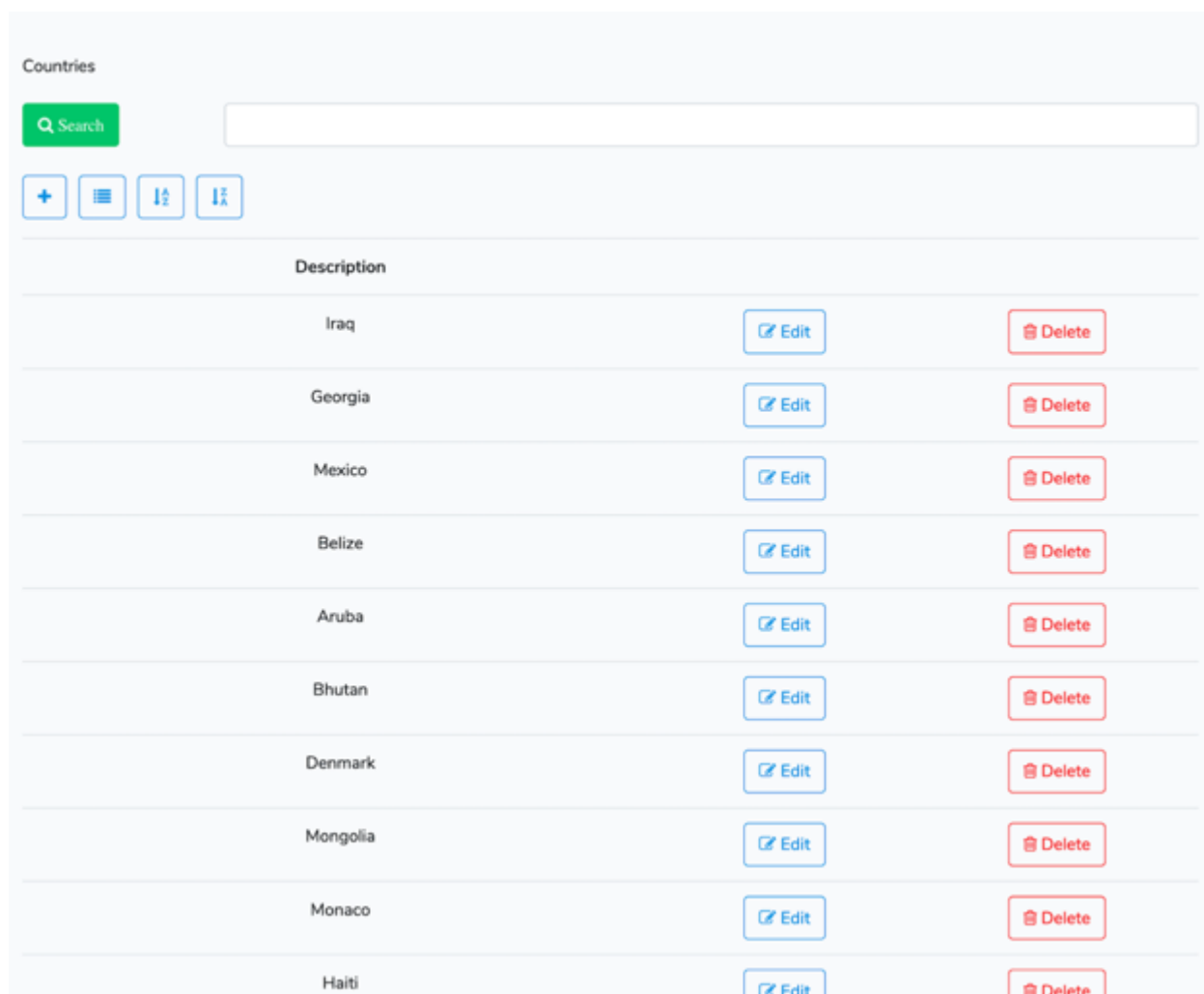


Description

Рисунок 4.56 – Корегування одиниць вимірювань

Сторінка Countris (рис. 4.57) – база даних країн, яка використовується при заповненні анкети. Адміністратор виконує всі дії, що і на сторінці Unit.



Countries

Description		
Iraq	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Georgia	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Mexico	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Belize	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Aruba	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Bhutan	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Denmark	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Mongolia	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Monaco	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Haiti	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Рисунок 4.57 – Countris



Estimates of Work			
Title	Created on	Status	
I	Nov 7, 2020	In process	<input checked="" type="checkbox"/>

Estimates of Materials			
Title	Created on	Status	
al	Aug 18, 2020	In process	<input checked="" type="checkbox"/>
1est2	Aug 20, 2020		<input checked="" type="checkbox"/>

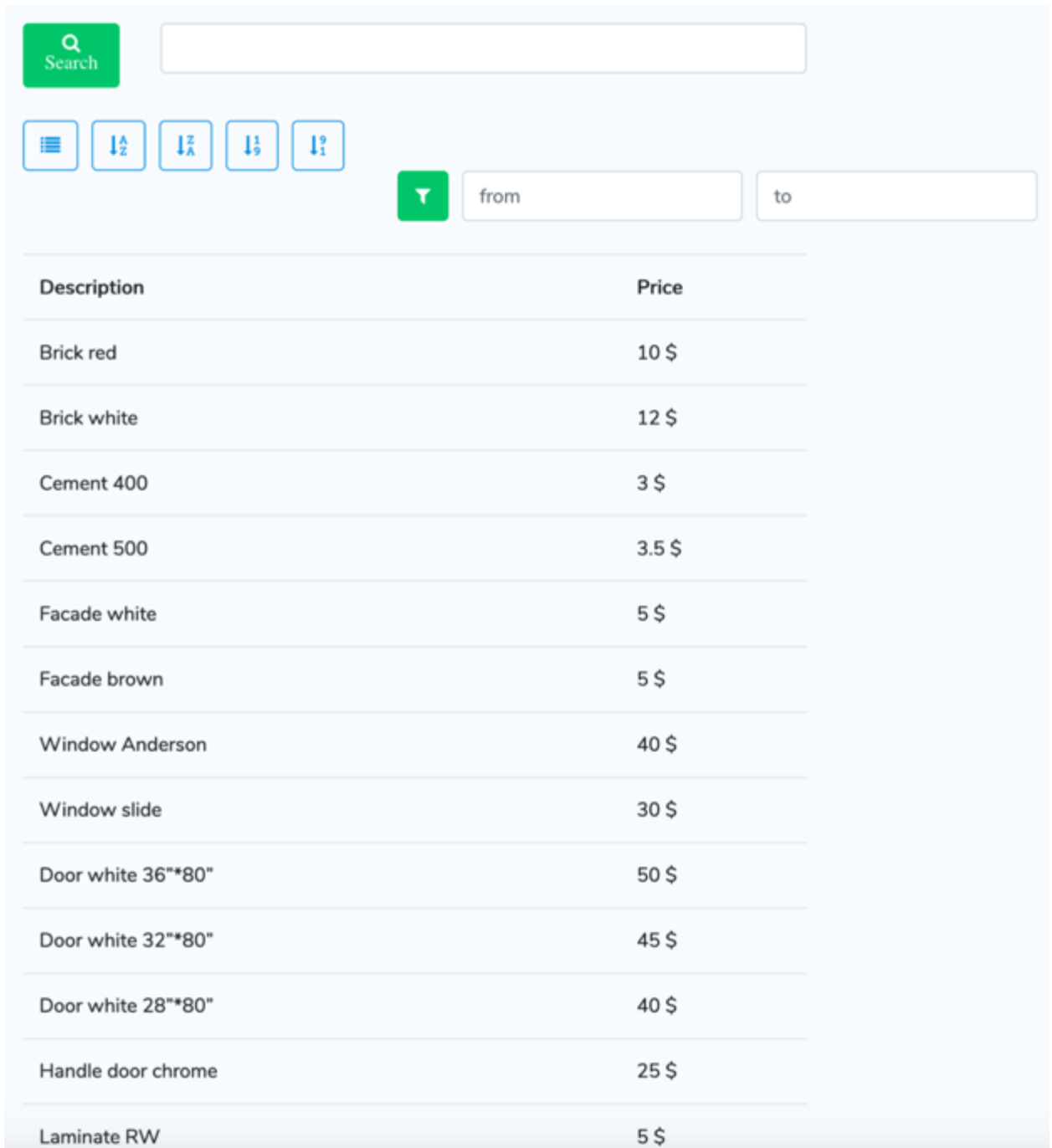
Рисунок 4.60 – Список кошторисів, які закріплені за одним проектом

Feedback – сторінка, на які клієнт залишає свій відгук (рис. 4.61)

Description

Рисунок 4.61 – Форма відгуку

Price of materials – інформаційно-ознайомлювальна сторінка цін на матеріали. Існують функція пошуку та фільтри (рис. 4.62).



The screenshot shows a web interface for a price list. At the top left is a green search button with a magnifying glass icon and the word "Search". To its right is a search input field. Below the search bar are five navigation icons: a list icon, and four icons representing different sorting or filtering options. To the right of these icons is a green button with a downward arrow, followed by two input fields labeled "from" and "to".

Description	Price
Brick red	10 \$
Brick white	12 \$
Cement 400	3 \$
Cement 500	3.5 \$
Facade white	5 \$
Facade brown	5 \$
Window Anderson	40 \$
Window slide	30 \$
Door white 36**80"	50 \$
Door white 32**80"	45 \$
Door white 28**80"	40 \$
Handle door chrome	25 \$
Laminate RW	5 \$

Рисунок 4.62 – Сторінка цін на матеріали

Profile Settings – вікно налаштування особистих даних клієнта (рис. 4.63).

Profile settings [← Back](#)

Name	<input type="text" value="Alena"/>
E-Mail Address	<input type="text" value="symskaja@yandex.ru"/>
Phone	<input type="text" value="0994850511"/>
Country	<input type="text" value="Ukraine"/>
Region	<input type="text" value="sumy"/>
City	<input type="text" value="sumy"/>
Street	<input type="text" value="romenskaja"/>
Zip code	<input type="text" value="12345"/>

Рисунок 4.63 – Profile Settings в аккаунті клієнта

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було проведено аналіз діяльності компанії замовника, визначено основні вимоги та потреби до розробки та сформовано мету і задачі для реалізації проекту.

Було досягнуто мету проекту, а саме розроблено веб-орієнтовану інформаційну систему формування кошторису будівельних робіт, яка дозволяє швидко та практично отримати оціночну документацію будівельних робіт на місці складання проекту, обговорюючи всі важливі моменти з клієнтом, а також дозволить систематизувати дані кошториси, що полегшуватиме звітність перед податковою інспекцією.

Проведено дослідження предметної області. Розглянуті аналоги програмного забезпечення для розробки кошторису будівельних робіт. Визначено функціональні вимоги, проведено планування, створено структуру робіт проекту, складено календарний план, проаналізовано ризики проекту.

Проведено структурно-функціональне моделювання, створено діаграму IDEF0, її декомпозицію та діаграму варіантів використання.

Проведено огляд та аналіз існуючих технологій розробки веб-орієнтованої інформаційної системи. Для реалізації було обрано інтегроване середовище розробки PhpStorm, гіпертекстову мову програмування HTML для розмітки веб-сторінок, Vue.js для створення інтерфейсу, таблиці стилів CSS для оформлення дизайну, мова JavaScript для анімації об'єктів, PHP-фреймворк Laravel 7 для опису серверної частини, MySQL як базу даних для підтримки роботи веб-сайту.

Спроектовано архітектуру розробки на основі шаблону MVC, створена база даних, розроблена серверна частина, яка відповідає за обробку запитів і реалізацію функціоналу. При розробці використовувався механізм міграції, що скоротило час розробки.

Результатом кваліфікаційної роботи магістра є готова веб-орієнтована інформаційна система для формування кошторису будівельних робіт. В майбутньому інформаційна система може бути вдосконалена та оснащена додатковими функціями, такими як складання звітності про податки на рік, відправлення повідомлень між користувачами, оновлення дизайну.

Створений програмний продукт буде застосовано в роботі DLDM компанії.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Применение IT-технологий в строительстве: сегодня и завтра, [Электронный ресурс] – режим доступа: <https://cyberleninka.ru/article/n/primenenie-it-tehnologiy-v-stroitelstve-segodnya-i-zavtra/viewer>
2. Информационные технологии в строительстве: описание и виды, применение на практике, [Электронный ресурс] – режим доступа: <http://fewdays.ru/repair-design/informacionnye-tehnologii-v-stroitelstve-opisanie-i-vidy-primenenie-na-praktike.html>
3. Информационные технологии в строительстве, [Электронный ресурс] – режим доступа: [https://spravochnick.ru/informacionnye\\_tehnologii/informacionnye\\_tehnologii\\_v\\_stroitelstve/](https://spravochnick.ru/informacionnye_tehnologii/informacionnye_tehnologii_v_stroitelstve/)
4. IT у сфері будівництва, [Электронный ресурс] – режим доступа: <http://yurholding.com/news/117-t-u-sfer-budvnictva.html>
5. Информационные системы строительных организаций: моделирование и оценка потребительского качества, [Электронный ресурс] – режим доступа: <https://rsue.ru/avtoref/Kudinov.pdf>
6. Информационные технологии в строительстве, Шумский А.Ю., [Электронный ресурс] – режим доступа: [https://rep.bntu.by/bitstream/handle/data/36353/Informacionnye\\_tekhnologii\\_v\\_stroitelstve.pdf?sequence=1](https://rep.bntu.by/bitstream/handle/data/36353/Informacionnye_tekhnologii_v_stroitelstve.pdf?sequence=1)
7. Информационные технологии в архитектуре и строительстве, Г.В. Прохорский, [Электронный ресурс] – режим доступа: <https://cdn1.ozone.ru/multimedia/1023960376.pdf>
8. Інтелектуальна архітектурна сапр archicad, [Электронный ресурс] – режим доступа: <https://books.googleusercontent.com/books/content?req=AKW5QadEZpqbz54If>



iatfKJK5hGnLP5BshC0aSEjTxMn9RsE\_JzNQjkGXjDpuF8uLRNvH2yCuysZ9  
yRL60AE2t245Y7zFrGuiSusDFDHLf2JHzgV7I37Z33\_HON\_CHOwn1rDSP2  
oS3C5CTjxoA7x\_wszaKwXDTqjS1CnK-3zciyuWvapd3E2c5p9zkGG-  
24r6ZoZrpC0koJexba0zHxB6ZqepO0qJihCMjBMzyGSlsZqAbYDThjHW27C  
W0B8qwIVB10g9D3zhPrEAJAsprBIQMAyFEvbjs9\_nOQ6PRpanTet25lpniou5  
zY

9. Archicad – GRAPHISOFT, [Электронный ресурс] – режим доступа: <https://graphisoft.com/solutions/products/archicad>
10. Применение IT-технологий в строительстве и архитектуре, [Электронный ресурс] – режим доступа: [https://www.karma-group.ru/construction\\_architecture/](https://www.karma-group.ru/construction_architecture/)
11. What is BIM (Building Information Modeling) , [Электронный ресурс] – режим доступа: <https://constructible.trimble.com/construction-industry/what-is-bim-building-information-modeling>
12. The Scope and Role of Information Technology in Construction, [Электронный ресурс] – режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.544.4090&rep=rep1&type=pdf>
13. BIM-проекування в будівництві, [Электронный ресурс] – режим доступа: <https://vs-pro.com.ua/ua/no-category/tehnologija-informacionnogo-modelirovanija-stroitelstva/>
14. Планування та проектування інформаційних систем, [Электронный ресурс] – режим доступа: [http://www.dut.edu.ua/uploads/1\\_842\\_23775847.pdf](http://www.dut.edu.ua/uploads/1_842_23775847.pdf)
15. Cost Estimating Guideline, [Электронный ресурс] – режим доступа: [https://www.state.nj.us/transportation/capital/pd/documents/Cost\\_Estimating\\_Guideline.pdf](https://www.state.nj.us/transportation/capital/pd/documents/Cost_Estimating_Guideline.pdf)
16. Исследовательская работа по теме "Информационные технологии в строительстве", [Электронный ресурс] – режим доступа: <https://megatalant.com/biblioteka/issledovatelskaya-rabota-po-teme-informacionnyye-tehnologii-v-stroitelstve-86356.html>

17. Система прозорого будівництва м. Сеул для ефективного та прозорого державного управління будівництвом, [Електронний ресурс] – режим доступу:  
<https://www.undp.org/content/dam/uspc/docs/USPC%20CCS%20Ukrainian.pdf>
18. Інформаційні системи і технології в бухгалтерському обліку, монографія, М.М. Бенько, [Електронний ресурс] – режим доступу:  
<https://knute.edu.ua/file/MTc=/00fe89dcf255176477f44d6060ac7347.pdf>
19. Інформаційні системи бухгалтерського обліку : підручник [для студ. вищ. навч. закл. спеціальності 7.050106 «Облік і аудит»] / Ф.Ф. Бутинець, С.В. Івахненко, Т.В. Давидюк, Т.В. Шахрайчук; заред. проф. Ф.Ф. Бутиця. – Житомир : ПП «Рута», 2002. – 544 с.
20. МСмета – кошторисна документація в АВК-5, [Електронний ресурс] – режим доступу: [https://msmeta.com.ua/bjudzhetnyj\\_koshtorys.php](https://msmeta.com.ua/bjudzhetnyj_koshtorys.php)
21. Автоматизація складання та розрахунку кошторисів в будівельній галузі, Кошторис 3 TQMsystems, [Електронний ресурс] – режим доступу:  
<https://tqm.com.ua/ua/1s-avtomatyzatsiia-haluzi-industry/1c-koshtorys-3>
22. Top 5 Technology Trends in Construction for 2020, [Електронний ресурс] – режим доступу: <https://homebusinessmag.com/management/technology-management/top-5-technology-trends-construction-2020/>
23. Coconstruct, [Електронний ресурс] – режим доступу:  
[https://www.coconstruct.com/u/construction-software?utm\\_medium=cpc&utm\\_campaign=Construction-Management&utm\\_source=GetApp](https://www.coconstruct.com/u/construction-software?utm_medium=cpc&utm_campaign=Construction-Management&utm_source=GetApp)
24. STACK, [Електронний ресурс] – режим доступу: <https://www.stackct.com/>
25. Acumatica ERP system – Accelerate your business!, [Електронний ресурс] – режим доступу: <https://www.acumatica.com/>
26. Acumatica Pricing, [Електронний ресурс] – режим доступу:  
<https://www.loganconsulting.com/acumatica/pricing/>

27. Make better spending decisions with real-time construction budgeting software, [Электронный ресурс] – режим доступа: <https://www.procore.com/project-financials/budget>
28. Comparison with Other Frameworks, [Электронный ресурс] – режим доступа: <https://vuejs.org/v2/guide/comparison.html>
29. Vue.js. Getting Started, [Электронный ресурс] – режим доступа: <https://012.vuejs.org/guide/>
30. Vue.js: особенности, применение и отличия от других Javascript фреймворков, [Электронный ресурс] – режим доступа: <https://stfalcon.com/ru/blog/post/vue-js-guide-to-tech>
31. React или Angular или Vue.js — что выбрать?, [Электронный ресурс] – режим доступа: <https://habr.com/ru/post/476312/>
32. 6 Reasons Why Vue.js Is Important For Web Development, [Электронный ресурс] – режим доступа: [https://medium.com/@shashank.nautiyal\\_20242/6-reasons-why-vue-js-is-important-for-web-development-b05ff3f8dee7](https://medium.com/@shashank.nautiyal_20242/6-reasons-why-vue-js-is-important-for-web-development-b05ff3f8dee7)
33. 10 Popular PHP frameworks in 2020, [Электронный ресурс] – режим доступа: <https://raygun.com/blog/top-php-frameworks/>
34. Why Laravel is the Best PHP Framework In 2020?, [Электронный ресурс] – режим доступа: <https://www.esparkinfo.com/why-laravel-is-the-best-php-framework.html>
35. 17 преимуществ использования Laravel в IT-индустрии, [Электронный ресурс] – режим доступа: <https://wezom.com.ua/blog/17-preimuschestv-ispolzovanija-laravel-v-it-industrii>
36. 12 Reasons Why Laravel Is The Best PHP Framework?, [Электронный ресурс] – режим доступа: <https://www.pontikis.net/blog/12-reasons-laravel-is-the-best-php-framework>
37. Ruby on Rails vs Django vs Laravel: The Ultimate Comparison of Popular Web Frameworks, [Электронный ресурс] – режим доступа: <https://steelkiwi.com/blog/ruby-django-laravel-frameworks-comparison/>

38. Создание функциональной модели информационной системы с помощью CASE-средства CA ERwin Process Modeler 7.3, [Электронный ресурс] – режим доступа: <http://window.edu.ru/resource/658/72658/files/stup552.pdf>
39. Методология IDEF0, [Электронный ресурс] – режим доступа: [http://e-learning.bmstu.ru/moodle/pluginfile.php/5193/mod\\_resource/content/1/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%20IDEF0.pdf](http://e-learning.bmstu.ru/moodle/pluginfile.php/5193/mod_resource/content/1/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%20IDEF0.pdf)
40. Моделирование программного обеспечения, [Электронный ресурс] – режим доступа:  
<http://elartu.tntu.edu.ua/bitstream/123456789/17796/1/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F.pdf>
41. Застосування UML для моделювання та проектування інформаційних систем, [Електронний ресурс] – режим доступу:  
<http://ir.stu.cn.ua/bitstream/handle/123456789/16930/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20uml%20%D0%B4%D0%BB%D1%8F%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%82%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D1%85%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.%20.pdf?sequence=1&isAllowed=y>
42. Бази даних та інформаційні системи, [Електронний ресурс] – режим доступу:

<http://ep3.nuwm.edu.ua/9129/3/%D0%A5%D0%B0%D1%80%D1%96%D0%B2%20%D0%9D.%D0%9E.pdf>

43. Бази даних: проектування та використання для обліку нерухомого майна, [Електронний ресурс] – режим доступу: [http://eprints.kname.edu.ua/41425/1/5%D0%9D%20%20%D0%91%D0%94\\_%D0%9D%D0%90%D0%B2%D1%87%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD.pdf](http://eprints.kname.edu.ua/41425/1/5%D0%9D%20%20%D0%91%D0%94_%D0%9D%D0%90%D0%B2%D1%87%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD.pdf)
44. Системи баз даних. Організація та проектування, [Електронний ресурс] – режим доступу: [http://eir.zp.edu.ua/bitstream/123456789/1738/1/Korniienko\\_System\\_Database.pdf](http://eir.zp.edu.ua/bitstream/123456789/1738/1/Korniienko_System_Database.pdf)
45. How to organize your project with PHP and Laravel to get the best structure in MVC pattern, [Електронний ресурс] – режим доступу: <https://steemit.com/php/@linko-crypto/how-to-organize-your-project-with-php-and-laravel-to-get-the-best-structure-in-mvc-pattern>
46. The basic architecture of Laravel applications, [Електронний ресурс] – режим доступу: [https://subscription.packtpub.com/book/web\\_development/9781788833912/1/ch01lv11sec11/the-basic-architecture-of-laravel-applications](https://subscription.packtpub.com/book/web_development/9781788833912/1/ch01lv11sec11/the-basic-architecture-of-laravel-applications)

**Додаток А**  
**Планування робіт**  
**Ідентифікація мети ІТ-проекту**

Для визначення деталізації мети проекту було використано технологію SMART. Це сучасний підхід до постановки працюючих цілей. Система постановки SMART-цілей дозволяє на етапі визначення мети узагальнити всю наявну інформацію, встановити прийнятні терміни роботи, визначити достатність ресурсів, надати ясні, точні, конкретні завдання. Кожна буква абрєвіатури SMART означає критерій ефективності поставлених цілей. Розглянемо кожен критерій SMART мети більш докладно.

У табл. А.1 відображається деталізація мети проекту дипломної роботи по технології SMART.

Таблиця А.1 – Деталізація мети методом SMART

Specific (Конкретна)	Розробка Веб-орієнтованої інформаційною системою формування кошторису будівельних робіт.
Measurable (Вимірювана)	Результативність вимірюється показником швидкості створення документації оціночної вартості робіт та матеріалів для відповідного проекту.
Agreed Upon (Досяжна)	Інформаційна система реалізується за допомогою HTML, CSS (web-сайт), PHP, Laravel (серверна частина), JS, Vue.js (клієнтська частина) та MySQL (база даних).
Realistic (Реалістична)	Для розробки проекту є всі програмні та технічні засоби, доступ до інтернет.
Time-related (Обмежена в часі)	Для створення інформаційної системи сформовано календарний план, так як є обмеження в часі.

## Планування змісту структури робіт IT-проекту

WBS (Work Breakdown Structure) – це узгоджена з результатами поставки ієрархічна декомпозиція робіт, які необхідно виконати для досягнення цілей проекту і створення обумовлених результатів проекту. Основна мета – визначити і структурувати весь зміст проекту. Іншими словами WBS містить всі роботи, які заплановані в проекті. Переваги декомпозиції:

- ефективний спосіб наочно відобразити весь обсяг проекту;
- фокусує увагу не на процесі, а на очікуваному результаті;
- забезпечує єдине розуміння результатів проекту та його обсягу, як зі сторони розробникам так і замовника;
- наочно обґрунтувати необхідність в ресурсах;
- допомагає запобігти ризики і зміни або принаймні знизити їх ймовірність та вплив;
- визначити і погодити контрольні точки проекту.

Тому для ефективної розробки проекту було розроблено WBS-діаграму, яка представлена на рис. А.1.

Основа планування контролю за проектом включає не лише визначення того, що таке робота, але встановлення загальних підходів до того, як буде виконуватися робота. Доповненням до WBS є організаційна структура розподілу (OBS), яка забезпечить ресурси, необхідні для виконання проектних робіт. Інтеграція організації з WBS гарантує, що вся робота над проектами враховується і що кожен елемент роботи призначається на рівень відповідальності, необхідний для планування, відстеження прогресу, накопичення витрат та звітності. Тому після складання WBS-діаграми необхідно оформити OBS-діаграму, завдяки чому відбудеться організація процесу розробки проекту через розподілу обов'язків між учасниками проекту. На рис. А.2 представлена OBS-діаграма проекту.

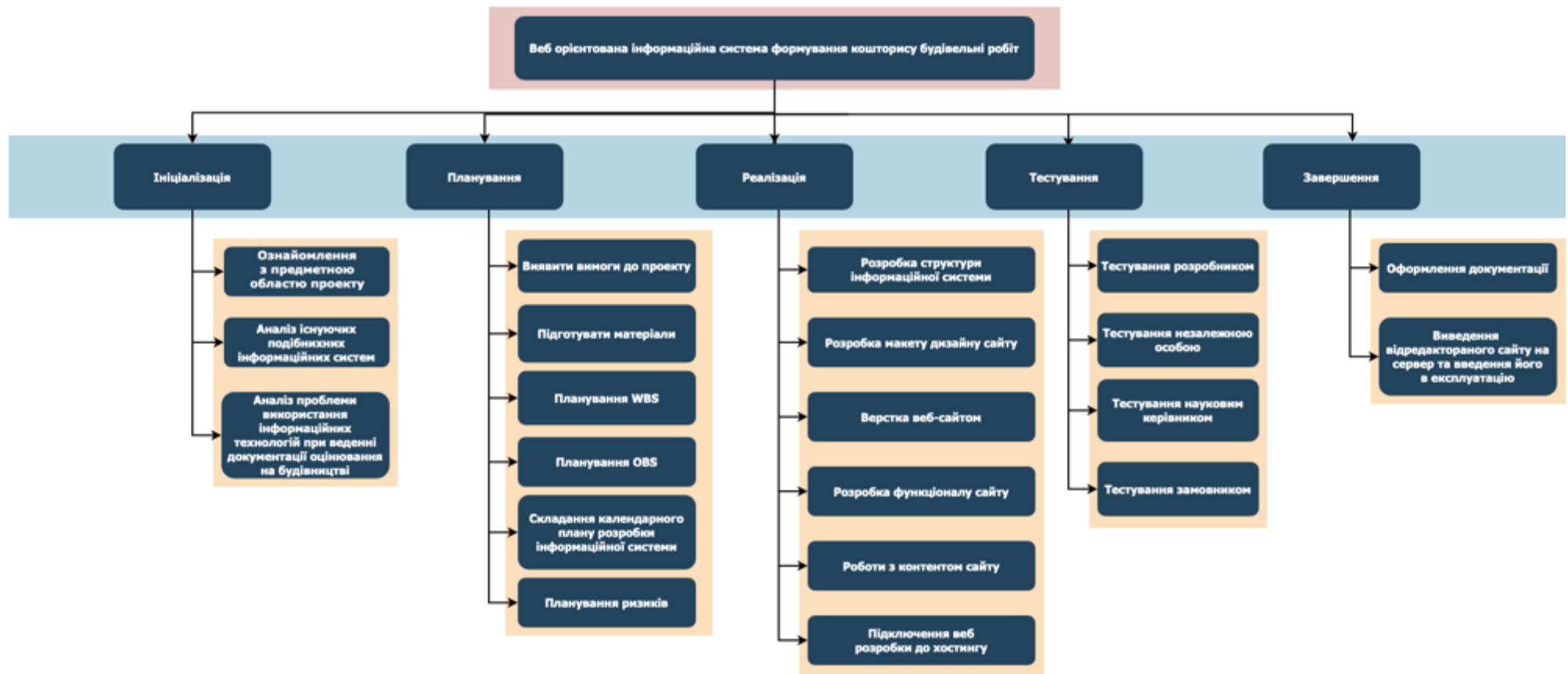


Рисунок А.1 — WBS-діаграма проекту «Веб-орієнтовна інформаційна система формування кошторису будівельних робіт»



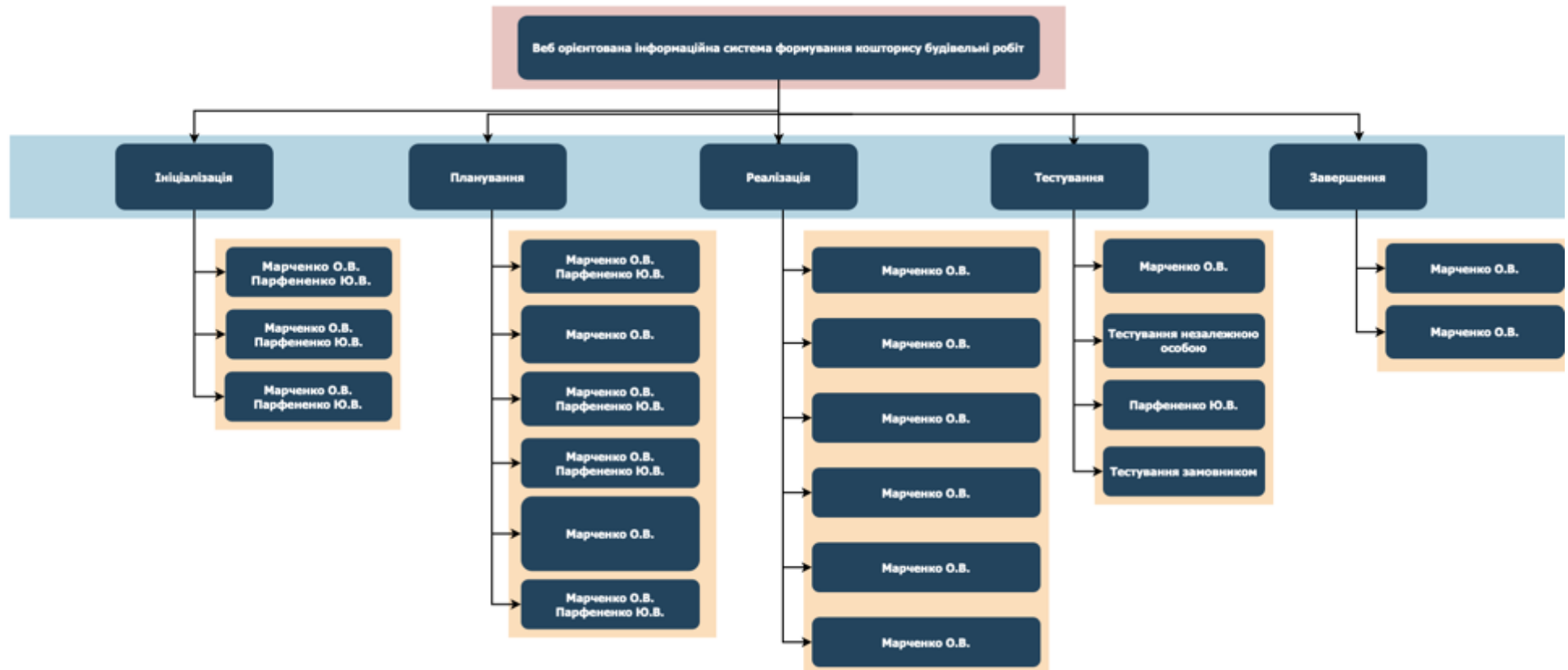


Рисунок А.2 – OBS-діаграма проекту «Веб-орієнтовна інформаційна система формування кошторису будівельних робіт»

## Побудова календарного графіку виконання ІТ – проекту

Календарне планування проекту полягає у визначенні і коригуванні календарних термінів виконання проекту в цілому і окремих його робіт, ілюстрації взаємозв'язків між окремими елементами проекту, оптимізації часового розподілу його ресурсного забезпечення. Інструментом календарного планування є календарні графіки. Календарні графіки відіграють важливу роль в процесі контролю за ходом реалізації проекту, оскільки вони поряд із плановими показниками відображають фактичне їх виконання.

У даний час широкого застосування набули (особливо в комп'ютерних програмах) саме модифіковані діаграми Ганта. Тому на базі діаграм WBS та OBS використаємо діаграму Ганта, яка відображає взаємозалежності між роботами і резервами часу.

На рис. А.3 представлена діаграма Ганта.

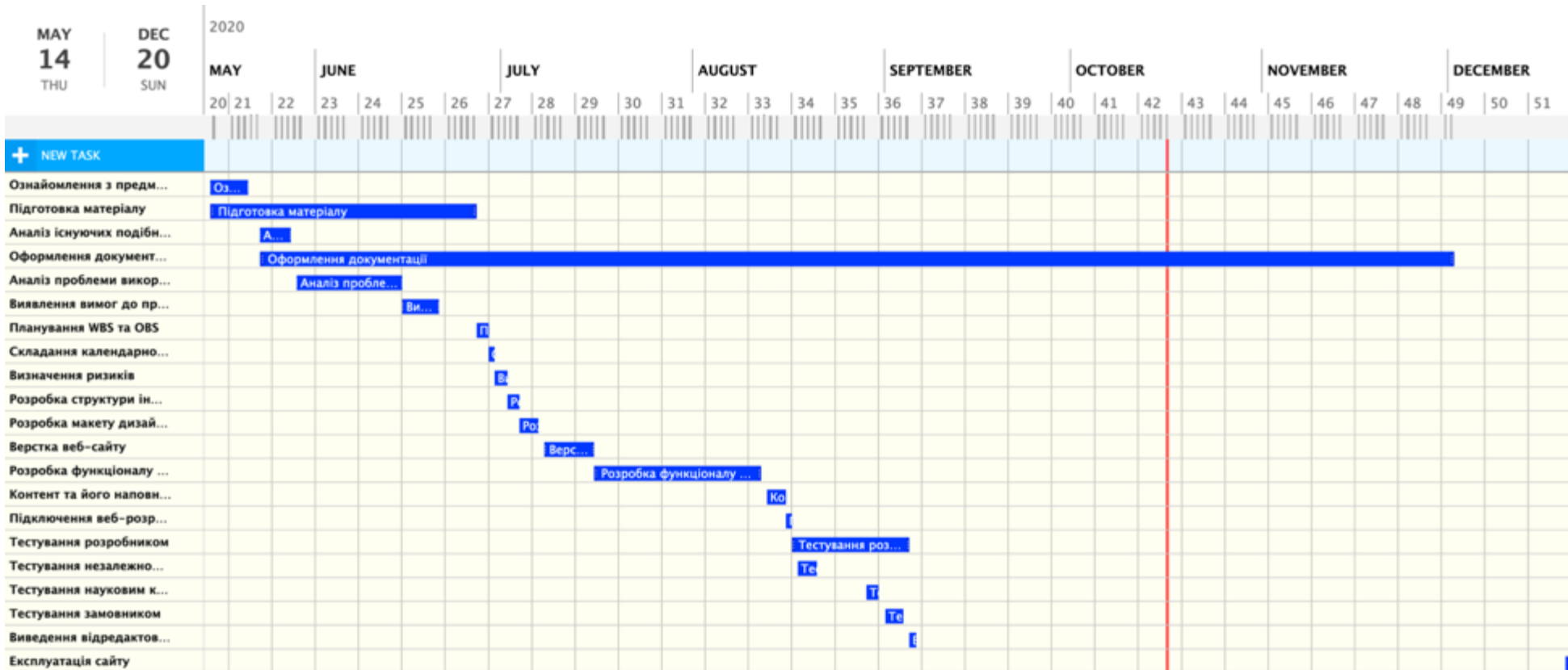


Рисунок А.3 – Діаграма Ганта проекту «Веб-орієнтовна інформаційна система формування кошторису будівельних робіт»

## Планування ризиків проекту

Відомо, що значна кількість проектів у галузі ІТ не відповідає поставленим цілям, бюджету або термінам – в середньому в світі цей показник перевищує 50%, а у державному секторі майже 70%. Найчастіше ці проблеми пов’язані з нестачею або взагалі відсутністю повного та якісного планування ризиками.

Аналізуючи процес розробки проекту: “Веб-орієнтована інформаційна система формування кошторису будівельних робіт”, було сформовано можливі ризики впливу на якість та ефективність кінцевого продукту (рис.А.4).

Ризики	Приклади	Ступінь ймовірності	Ступінь впливу	Рішення
Ризик появи нових вимог до проекту	-нові задачі -нові функції, що приводить до значних повторних робіт із проєктування системи	Низький	Низький	- уточнити вимоги до створення проекту з замовником
Збільшення зовнішнього навантаження на виконання проєкту внаслідок виникнення інших пріоритетних завдань.	- допрацювання попереднього проєкту	Низький	Середній	- планування розробки інформаційної системи з екстремним часом
Технологічні ризики на етапі завершення	-несправність обладнання -неможливість виставити проєкт на хостинг (безкоштовний) -непідтримання хостингом деяких модулів	Низький	Середній	- наявність резервного обладнання - попередження замовника про обмеження безкоштовних хостингів
Ризик залучення інших спеціалістів при створенні проєкту	-хвороба розробника	Низький	Середній	- залучення кваліфікованих експертів
Виявлення помилок у плануванні та проєктуванні	-виявили помилку при функціональному тестуванні	Середній	Середній	-здійснювати проміжні перевірки

Рисунок А.4 – Таблиця ризиків проекту

## Додаток Б

Файл EstimateMaterialController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\EstimateMaterial;
use App\Material;
use App\MaterialForEstimate;
use App\Project;
use App\TypeWork;
use App\Unit;
use App\User;
use App\Work;
use Barryvdh\DomPDF\Facade as PDF;
use Illuminate\Http\Request;
```

```
use App\Http\Requests\ProjectRequest;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Mail;
```

```
class EstimateMaterialController extends Controller
{
```

```
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Project $project)
    {
        $estimates_material = EstimateMaterial::where('id_project',$project->id)->get();
        return view('estimates_materials.estimatesMaterialIndex',
compact('estimates_material'),[
            'id_project' => $project->id
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
```

```

* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\RedirectResponse|\Illuminate\Routing\Redirector
*/
public function store(Request $request)
{
    $estimate_material = new EstimateMaterial([
        'name' => $request->get('Name'),
        'id_customer' => $request->get('selectedOptionCustomer'),
        'id_project' => $request->get('idProject'),
        'total_price' => $request->get('total_price'),
        'status' => 'New',
    ]);
    $estimate_material->save();

    $last_estimate = EstimateMaterial::latest()->first();

    foreach ($request->materials as $key => $value) {
        $material = new MaterialForEstimate([
            'id_material' => $value['id_material'],
            'id_estimate' => $last_estimate->id,
            'id_unit' => $value['id_unit'],
            'quantity' => $value['quantity'],
            'price' => $value['price'],
            'total_amount' => $value['total_amount'],
        ]);

        $material->save();
    }
    return redirect('/projects')->with('success', 'Stock has been added');
}

/**
 * Display the specified resource.
 *
 * @param $id
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show($id)
{
    return view('estimates_materials.show', compact('id'),
        [

```

```

        'customers' => User::select('id','Name')->where('regType',0)->get()-
>toArray(),
        'units' => Unit::select('id','Name')->get()->toArray(),
        'materials_all' => Material::select('id','Name','price')->get()->toArray(),
    ]);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function edit($id)
{
    return view('estimates_materials.edit', compact('id'),
        [
            'id_project' => EstimateMaterial::select('id_project')->where('id',$id)-
>first(),
            'customers' => User::select('id','Name')->where('regType',0)->get()-
>toArray(),
            'units' => Unit::select('id','Name')->get()->toArray(),
            'materials_all' => Material::select('id','Name','price')->get()->toArray(),
        ]);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, EstimateMaterial $estimates_material)
{
    $estimates_material->name = $request->Name;
    $estimates_material->id_customer = $request->selectedOptionCustomer;
    $estimates_material->id_project = $request->idProject;
    $estimates_material->total_price = $request->total_price;
    $estimates_material->save();
}

```

```

    $array_materials          =          MaterialForEstimate::select('id')-
>where('id_estimate',$estimates_material->id)->get()->toArray();
    $stack = array();

    foreach ($request->materials as $key => $value) {
        if($value['id'] == 0){
            $material = new MaterialForEstimate([
                'id_material' => $value['id_material'],
                'id_estimate' => $value['id_estimate'],
                'id_unit' => $value['id_unit'],
                'quantity' => $value['quantity'],
                'price' => $value['price'],
                'total_amount' => $value['total_amount'],
            ]);
        }
        else{
            $material = MaterialForEstimate::where('id', $value['id'])->first();
            $material->id_material = $value['id_material'];
            $material->id_estimate = $value['id_estimate'];
            $material->quantity = $value['quantity'];
            $material->price = $value['price'];
            $material->total_amount = $value['total_amount'];
        }
        $material->save();

        array_push($stack, $value['id']);
    }

    foreach ($array_materials as $keya => $valuea) {
        $result = array_search($valuea['id'], $stack);

        if ($result === false){
            $material = MaterialForEstimate::where('id', $valuea['id'])->first();
            $material->delete();
        }
    }
}

/**
 * Remove the specified resource from storage.
 *
 * @param Request $request
 * @param EstimateMaterial $estimates_material
 * @return \Illuminate\Http\Response

```



```

* @throws \Exception
*/
public function destroy(Request $request, EstimateMaterial $estimates_material)
{

    $estimates_material->delete();
    return redirect('/projects');

}

/**
 * Получение данных EstimateMaterial
 *
 * @param $id_estimate_material
 * @return array
 */
public function getEstimateMaterialInfo($id_estimate_material)
{

    $estimate_materials = EstimateMaterial::where('id',$id_estimate_material)-
>first();

    $materials = MaterialForEstimate::where('id_estimate',$id_estimate_material)-
>get()->toArray();

    return ['estimate_materials' => $estimate_materials, 'materials' => $materials];

}

public function download($id_estimate_materials)
{
    if($id_estimate_materials == 0){
        $last_estimate = EstimateMaterial::latest()->first();
        $id_estimate_materials = $last_estimate->id;
    }
    $estimate_materials = EstimateMaterial::where('id',$id_estimate_materials)-
>first();

    $materials = MaterialForEstimate::select(
        'materials_for_estimate.quantity as quantity',
        'materials_for_estimate.total_amount as total_amount',
        'materials.Name AS material_name',
        'units.Name AS unit_name'
    )

```

```

->leftJoin('materials', 'materials.id', '=', 'materials_for_estimate.id_material')
->leftJoin('units', 'units.id', '=', 'materials_for_estimate.id_unit')
->where('materials_for_estimate.id_estimate',$id_estimate_materials)
->get()
->toArray();

```

```

$customer = User::select(
    'countries.name as country',
    'users.name',
    'users.email',
    'users.phone',
    'users.region',
    'users.city',
    'users.street',
    'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',$estimate_materials->id_customer)
->first();

```

```

$constructor = User::select(
    'countries.name as country',
    'users.name',
    'users.email',
    'users.phone',
    'users.region',
    'users.city',
    'users.street',
    'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',\Auth::id())
->first();

```

```

$pdf = PDF::loadView('estimates_materials.pdf', [
    'estimate_materials' => $estimate_materials,
    'customer' => $customer,
    'constructor' => $constructor,
    'materials' => $materials]);

```

```

return $pdf->download('estimates_materials.pdf');
}

```

```

public function sendEmail($id_estimate_materials){

```

```
$estimate_materials = EstimateMaterial::where('id',$id_estimate_materials)->first();
```

```
$materials = MaterialForEstimate::select(
    'materials_for_estimate.quantity as quantity',
    'materials_for_estimate.total_amount as total_amount',
    'materials.Name AS material_name',
    'units.Name AS unit_name'
)
->leftJoin('materials', 'materials.id', '=', 'materials_for_estimate.id_material')
->leftJoin('units', 'units.id', '=', 'materials_for_estimate.id_unit')
->where('materials_for_estimate.id_estimate',$id_estimate_materials)
->get()
->toArray();
```

```
$customer = User::select(
    'countries.name as country',
    'users.name',
    'users.email',
    'users.phone',
    'users.region',
    'users.city',
    'users.street',
    'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',$estimate_materials->id_customer)
->first();
```

```
$constructor = User::select(
    'countries.name as country',
    'users.name',
    'users.email',
    'users.phone',
    'users.region',
    'users.city',
    'users.street',
    'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',\Auth::id())
->first();
```

```

$pdf = PDF::loadView('estimates_materials.pdf', [
    'estimate_materials' => $estimate_materials,
    'customer' => $customer,
    'constructor' => $constructor,
    'materials' => $materials]);

$data["email"]=$customer->email;
$data["client_name"]=$customer->name;

try{
    Mail::send('mails.mail_estimate_material',           $data,
function($message)use($data,$pdf) {
    $message->to($data["email"], $data["client_name"])
        ->subject('Estimates')
        ->attachData($pdf->output(), "estimates_materials.pdf");
    });
}catch(JWTException $exception){
    $this->serverstatusCode = "0";
    $this->serverstatusdes = $exception->getMessage();
}
return redirect('/home');
}

public function setFinishStatus($id_estimate_materials)
{
    $estimate_materials = EstimateMaterial::where('id',$id_estimate_materials)-
>first();

    $estimate_materials->status = 'Done';
    $estimate_materials->save();

    $user =\Auth::user();
    $projects = Project::select(
        'projects.id',
        'projects.name',
        'projects.created_at',
        'users.name as customer_name'
    )
    ->leftJoin('users', 'users.id', '=', 'projects.id_customer')
    ->where('id_user',$user->id)->get();
return view('projects.index', compact('projects'));
}

```

```
public function setProcessStatus($id_estimate_materials){

    $estimate_materials = EstimateMaterial::where('id',$id_estimate_materials)-
>first();

    $estimate_materials->status = 'In process';
    $estimate_materials->save();

    $user =\Auth::user();

    return view('customers.my_projects',
        [
            'user_info' => User::where('id',$user->id)->get()->first(),
            'projects' => Project::where('id_customer',$user->id)->get(),
        ]);
    }
}
```

Файл EstimateWorkController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\EstimateWork;
use App\ExampleWork;
use App\Project;
use App\TypeWork;
use App\Unit;
use App\User;
use App\Work;
use Barryvdh\DomPDF\Facade as PDF;
use Illuminate\Support\Facades\URL;
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Mail;
```

```
class EstimateWorkController extends Controller
```

```
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Project $project)
    {
        $estimates_work = EstimateWork::where('id_project',$project->id)->get();
        return view('projects.estimatesWorkIndex', compact('estimates_work'),[
            'id_project' => $project->id
        ]);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\RedirectResponse|\Illuminate\Routing\Redirector
     */
}
```

```

public function store(Request $request)
{

    $estimate_work = new EstimateWork([
        'name' => $request->get('Name'),
        'id_customer' => $request->get('selectedOptionCustomer'),
        'id_project' => $request->get('idProject'),
        'total_price' => $request->get('total_price'),
        'status' => 'New',
    ]);
    $estimate_work->save();

    $last_estimate = EstimateWork::latest()->first();

    foreach ($request->works as $key => $value) {
        $work = new Work([
            'id_example_work' => $value['id_example_work'],
            'id_type_work' => $value['id_type_work'],
            'id_estimate' => $last_estimate->id,
            'id_unit' => $value['id_unit'],
            'quantity' => $value['quantity'],
            'price' => $value['price'],
            'total_amount' => $value['total_amount'],
        ]);

        $work->save();
    }
    return redirect('/projects');

}

/**
 * Display the specified resource.
 *
 * @param EstimateWork $estimateWork
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show($id)
{
    return view('estimates_work.show', compact('id'),
        [
            'customers' => User::select('id','Name')->where('regType',0)->get()->toArray(),
            'typeWorks' => TypeWork::select('id','Name')->get()->toArray(),

```

```

        'units' => Unit::select('id','Name')->get()->toArray(),
        'examples_all' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->get()->toArray(),

        'examples_exterior' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',1)
        ->get()->toArray(),

        'examples_interior' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',2)
        ->get()->toArray(),

        'examples_furniture' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',3)
        ->get()->toArray()
    });
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function edit($id)
{
    return view('estimates_work.edit', compact('id'),
        [
            'id_project' => EstimateWork::select('id_project')->where('id',$id)->first(),
            'customers' => User::select('id','Name')->where('regType',0)->get()-
>toArray(),

```



```

        'typeWorks' => TypeWork::select('id','Name')->get()->toArray(),
        'units' => Unit::select('id','Name')->get()->toArray(),
        'examples_all' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->get()->toArray(),

        'examples_exterior' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',1)
        ->get()->toArray(),

        'examples_interior' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',2)
        ->get()->toArray(),

        'examples_furniture' =>
ExampleWork::select('example_works.id','example_works.Name','example_works.pri
ce','units.id as id_unit_base')
        ->leftjoin('units','units.name','=', 'example_works.unit_base')
        ->where('id_type_work',3)
        ->get()->toArray()
    ];
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, EstimateWork $estimates_work)
{
    $estimates_work->name = $request->Name;
    $estimates_work->id_customer = $request->selectedOptionCustomer;

```

```

$estimates_work->id_project = $request->idProject;
$estimates_work->total_price = $request->total_price;
$estimates_work->save();

$array_works = Work::select('id')->where('id_estimate',$estimates_work->id)-
>get()->toArray();
$stack = array();

foreach ($request->works as $key => $value) {
    if($value['id'] == 0){
        $work = new Work([
            'id_example_work' => $value['id_example_work'],
            'id_type_work' => $value['id_type_work'],
            'id_estimate' => $value['id_estimate'],
            'id_unit' => $value['id_unit'],
            'quantity' => $value['quantity'],
            'price' => $value['price'],
            'total_amount' => $value['total_amount'],
        ]);
    }
    else{
        $work = Work::where('id', $value['id'])->first();
        $work->id_example_work = $value['id_example_work'];
        $work->id_estimate = $value['id_estimate'];
        $work->id_type_work = $value['id_type_work'];
        $work->quantity = $value['quantity'];
        $work->price = $value['price'];
        $work->total_amount = $value['total_amount'];
    }
    $work->save();

    array_push($stack, $value['id']);
}

foreach ($array_works as $keya => $valuea) {
    $result = array_search($valuea['id'], $stack);

    if ($result === false){
        $material = Work::where('id', $valuea['id'])->first();
        $material->delete();
    }
}
}

```

```

/**
 * Remove the specified resource from storage.
 *
 * @param Request $request
 * @param EstimateWork $estimates_work
 * @return \Illuminate\Http\Response
 * @throws \Exception
 */
public function destroy(Request $request, EstimateWork $estimates_work)
{
    $estimates_work->delete();
    return redirect('/projects');
}

/**
 * Получение данных EstimateWork
 *
 * @param $id_estimate_work
 * @return array
 */
public function getEstimateWorkInfo($id_estimate_work)
{
    $estimate_work = EstimateWork::where('id',$id_estimate_work)->first();

    $works = Work::where('id_estimate',$id_estimate_work)->get()->toArray();

    return ['estimate_work' => $estimate_work, 'works' => $works];
}

public function download($id_estimate_work)
{
    if($id_estimate_work == 0){
        $last_estimate = EstimateWork::latest()->first();
        $id_estimate_work = $last_estimate->id;
    }
    $estimate_work = EstimateWork::where('id',$id_estimate_work)->first();

    $works = Work::select(
        'works.quantity as quantity',

```

```

    'works.total_amount as total_amount',
    'example_works.Name AS work_name',
    'units.Name AS unit_name'
  )
  ->leftJoin('example_works', 'example_works.id', '=',
'works.id_example_work')
  ->leftJoin('units', 'units.id', '=', 'works.id_unit')
  ->where('works.id_estimate',$id_estimate_work)
  ->get()
  ->toArray();

```

```

$customer = User::select(
  'countries.name as country',
  'users.name',
  'users.email',
  'users.phone',
  'users.region',
  'users.city',
  'users.street',
  'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',$estimate_work->id_customer)
->first();

```

```

$constructor = User::select(
  'countries.name as country',
  'users.name',
  'users.email',
  'users.phone',
  'users.region',
  'users.city',
  'users.street',
  'users.zip'
)
->leftJoin('countries', 'countries.id', '=', 'users.id_country')
->where('users.id',\Auth::id())
->first();

```

```

$pdf = PDF::loadView('estimates_work.pdf', [
  'estimate_work' => $estimate_work,
  'customer' => $customer,
  'constructor' => $constructor,

```

```

    'works' => $works]);

return $pdf->download('estimates_work.pdf');
}

public function sendEmail($id_estimate_work){

    $estimate_work = EstimateWork::where('id',$id_estimate_work)->first();

    $works = Work::select(
        'works.quantity as quantity',
        'works.total_amount as total_amount',
        'example_works.Name AS work_name',
        'units.Name AS unit_name'
    )
    ->leftJoin('example_works', 'example_works.id', '=',
'works.id_example_work')
    ->leftJoin('units', 'units.id', '=', 'works.id_unit')
    ->where('works.id_estimate',$id_estimate_work)
    ->get()
    ->toArray();

    $customer = User::select(
        'countries.name as country',
        'users.name',
        'users.email',
        'users.phone',
        'users.region',
        'users.city',
        'users.street',
        'users.zip'
    )
    ->leftJoin('countries', 'countries.id', '=', 'users.id_country')
    ->where('users.id',$estimate_work->id_customer)
    ->first();

    $constructor = User::select(
        'countries.name as country',
        'users.name',
        'users.email',
        'users.phone',
        'users.region',
        'users.city',
        'users.street',

```

```

        'users.zip'
    )
    ->leftJoin('countries', 'countries.id', '=', 'users.id_country')
    ->where('users.id', \Auth::id())
    ->first();

$pdf = PDF::loadView('estimates_work.pdf', [
    'estimate_work' => $estimate_work,
    'customer' => $customer,
    'constructor' => $constructor,
    'works' => $works]);

$data["email"]=$customer->email;
$data["client_name"]=$customer->name;

try{
    Mail::send('mails.mail_estimate_work',                                     $data,
function($message)use($data,$pdf) {
    $message->to($data["email"], $data["client_name"])
        ->subject('Estimates')
        ->attachData($pdf->output(), "estimates_work.pdf");
    });
}catch(JWTException $exception){
    $this->serverstatusCode = "0";
    $this->serverstatusdes = $exception->getMessage();
}
return redirect('/home');
}

public function setFinishStatus($id_estimate_work){

    $estimate_work = EstimateWork::where('id',$id_estimate_work)->first();

    $estimate_work->status = 'Done';
    $estimate_work->save();

    $user =\Auth::user();
    $projects = Project::select(
        'projects.id',
        'projects.name',
        'projects.created_at',
        'users.name as customer_name'
    )
    ->leftJoin('users', 'users.id', '=', 'projects.id_customer')

```

```
->where('id_user',$user->id)->get();
return view('projects.index', compact('projects'));
}

public function setProcessStatus($id_estimate_work){

    $estimate_work = EstimateWork::where('id',$id_estimate_work)->first();

    $estimate_work->status = 'In process';
    $estimate_work->save();

    $user =\Auth::user();

    return view('customers.my_projects',
        [
            'user_info' => User::where('id',$user->id)->get()->first(),
            'projects' => Project::where('id_customer',$user->id)->get(),
        ]);
}
}
```