

Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Веб орієнтовна система оцінки собівартості товарів та послуг

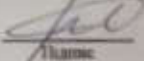
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-2  В.В. Дасиґа
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: с.т.н., доцент кафедри КНІТ  О.А. Пасічник
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: с.т.н., доцент кафедри КНІТ  Р.О. Базрії
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

08 червня 2021 р.



О.В. Бармак
Ініціали, прізвище

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій

(підпис)

д.т.н., професор О.В. Бармак

«08» лютого 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Веб орієнтовна система оцінки собівартості товарів та послуг»

2. Завдання видано студенту Дасизи Владиславу Вікторовичу

(прізвище, ім'я, по батькові)

3. Керівник роботи к.т.н., доцент кафедри КНІТ Пасічник Олександр Анатолійович

(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «05» лютого 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка веб орієнтовна система оцінки собівартості товарів та послуг. Для клієнта інформаційної системи необхідно реалізувати: систематичну оцінку товарів та послуг з урахуванням всіх можливих критеріїв.

Виконавець: студент 4 курсу, група КН-17-2
Курс, група виконавця

Підпис

В.В. Дасига
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада

Підпис

О.А. Пасічник
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Веб орієнтовна система оцінки собівартості товарів та послуг»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-2 Дасига Владислав Вікторович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КНІТ Пасічник Олександр Анатолійович

Кваліфікаційна робота бакалавра містить:

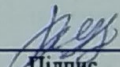
Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
55	18	1	20	3

Мета розробки кваліфікаційної роботи бакалавра, є веб застосунок для підрахунку собівартості товарів та послуг, який повинний забезпечити інформативність майбутніх клієнтів на етапі вирішення, який програмний продукт вони хочуть отримати.

Розроблений застосунок повністю готовий до використання, тому в подальшому для нього можна придбати хостинг та налагодити обслуговування для реальних задач. Напрямами практичного використання розробленої інформаційної системи визначено реалізацію застосунку для підрахунку собівартості продукту.

Ключові слова: сфера, застосунок, інформаційна система, розрахунок.

Виконавець: студент 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

В.В. Дасига
Ініціали, прізвище

Зміст

Вступ.....	4
Розділ 1	5
Характеристика предметної області і постановка задачі.....	5
1.1 Аналіз предметної області.....	5
1.2 Аналіз інформаційного забезпечення предметної області	6
1.3 Аналіз сучасних засобів створення програмного забезпечення	7
1.4 Постановка задачі та вимоги до розробки інформаційної системи.....	12
Розділ 2	14
Проектування інформаційної системи.....	14
2.1 Функціональна структура та бізнес-процеси системи	14
2.2 Інформаційна структура системи	22
2.3 Вибір засобів розробки інформаційної системи	32
2.3.1 Вибір мови програмування	32
2.3.2 Вибір бібліотеки	35
2.3.3 Вибір системи керування базами даних	37
2.3.4 Вибір засобу візуалізації	39
Розділ 3	40
Програмна реалізація	40
3.1 Структура і функціональне призначення модулів системи.....	40
3.2 Розробка програмних модулів	42
3.3 Тестування інформаційної системи	44
3.4 Інструкція користувача.....	46
3.5 Вимоги до розгортання інформаційної системи.....	52
Перелік посилань.....	54
Висновки	56

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ІС	Інформаційна система
БД	База даних
ІТ	Інформаційні технології
ПЗ	Програмне забезпечення
ПП	Програмний продукт
СКБД	Система керування базами даних
ТОВ	Товариство з обмеженою відповідальністю
СЕО	Chief executive officer
НВК	Навчально-виховні комплекси
JS	JavaScript
ООП	Об'єктно-орієнтоване програмування
CSS	Cascadical styles
HTML	Hyper text markup language

Вступ

В наш час про використання ІТ, у тому числі новітніх, незалежно від сфери застосування пишуть і говорять багато.[1]

Мета розробки кваліфікаційної роботи бакалавра, є веб застосунок для підрахунку собівартості товарів та послуг, який повинний забезпечити інформативність майбутніх клієнтів на етапі вирішення, який програмний продукт вони хочуть отримати.

Розробка буде виконуватися за допомогою бібліотеки React, з використанням не реляційної БД Firebase Realtime Database. [2]

Інформаційні технології - це інструмент, що відповідає інформаційну діяльність та її роботу, пов'язану з підготовкою та прийняттям рішень.

Технологіям процесів інформаційного забезпечення властиві організованість і впорядкованість, які протиставляються різними процесами процесам. Сам термін «технологія» виник у галузі матеріального виробництва. Інформаційні технології в цьому контексті можна вважати технологіями для використання програмно-апаратних засобів техніки для обчислень і телекомунікацій в конкретній сфері, або предметній області

Отже, звідси можна зробити висновок, що використання інформаційних технологій з метою покращення інформативного забезпечення клієнтів, покращить досвід співпраці з компанією, та принесе відкритість та розуміння формування цін даною компанією.

Розділ 1

Характеристика предметної області і постановка задачі

1.1 Аналіз предметної області

Вартість товару – уречевлена в товарі суспільно необхідна праця товаровиробників та виробничі відносини між людьми, що виникають у процесі виготовлення продукту внаслідок суспільного поділу праці й здійснюються між ними на основі витрат праці.

Закон вартості

Це закон виробництва, згідно з яким виробництво та обмін товарів повинні здійснюватися відповідно до суспільно необхідних витрат праці. Іншими словами, згідно з ним ціна повинна відповідати вартості товару або продукту.

Це такий економічний важіль. Коли його використовують, виробники починають регулювати ціну, домагаються її зниження або підвищення. Такий закон існує тільки тоді, коли є конкуренти. Адже якщо їх не буде, то не буде сенсу змінювати її в кращу або гіршу для людей сторону.

А тим самим знизиться і ринкова вартість товару, та й взагалі весь світ стане іншим, і економіка теж. Однак конкуренція є завжди і сподіватися на це марно. Якщо не застосовувати такий закон, то компанія може швидко збанкрутувати.

Вартість продукту – це ті ресурси і кошти, які витратили виробники на проект.

Продукт, тобто результат діяльності розглядається в якості одного з критеріїв творчості (а в контексті прагматичного підходу - як основний критерій). Продукт не обмежується рамками мистецтва і науки. Це може бути ринковий товар, сервіс, процес, технологія і т.д. «Незакінчена» творчість, не призводить до результату у вигляді перспективної ідеї, проекту, вирішення проблеми і т.д., в контексті прагматичного підходу не є творчістю.

Основні критерії вартості продукту:

—новизна - продукт повинен бути оригінальним і новим;

- корисність - продукт повинен вирішувати якусь проблему, служити визначеній меті;
- реалізація - продукт не повинен залишатися просто ідеєю, він повинен бути фактично проведений;
- час розробки продукту;
- функціонал та призначення;
- надійність.

На вартість продукту може впливати також багато інших факторів, але перш за все це все-таки складність реалізації.

1.2 Аналіз інформаційного забезпечення предметної області

Головна мета розрахунку вартості продукту є тим, що замовник онлайн може оцінити вартість свого замовлення без втрати свого часу на час очікування розрахунків. Оцінка вартості товарів і послуг здійснюється на основі їх споживчої вартості та залежності вартості від покупця.

Клієнт сам грає роль оцінювача в процесі оцінки. Для того, щоб оцінити вартість інтернет-товару, йому або їй потрібно врахувати всі витрати, які товар повинен сплатити за споживачів.

Однак це не означає, що оцінювач повинен дозволити ціні товару зростати після оцінки.

Кожна компанія або власник товару також відіграє важливу роль у визначенні ціни на свій товар.

Було розглянуто сайти з прорахунком вартості кінцевого продукту. В основному переваги таких це їх швидкість розрахунку вартості. Та розуміння замовника з чого складається ціна на продукт.

Основні недоліки полягають у тому, що у більшості таких калькуляторів розрахунку вартості продукту:

- не зручний інтерфейс;
- незрозумілі платні послуги;
- не розраховує кінцеву вартість продукту;
- мала кількість вибору функціоналу.

Для аналізу було обрано сайт розрахунку кінцевого продукту «LitsLink». Цей сайт несе в собі вирішення основних питань але має усі недоліки з перерахунку. Досліджуючи зразки з'ясувалось, що кожній фірмі потрібно конструювати свій калькулятор для розрахунку, так як усі фірми надають послуги різного рівня та у них працюють різної кваліфікації працівники.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Десктоп-додатки є додаток для настільного комп'ютера. Це програма, яка працює безпосередньо на компютері користувача.

Десктоп-додатки зручні для користувачів настільних комп'ютерів, такі програми зазвичай інтегровані з різними офісними та іншими настільними додатками.

Зазвичай розробка мобільних додатків ведеться на основі платформи .NET[3] Framework, технології Windows Forms, WPF (Windows Presentation Foundation), мова програмування C#, патернів асинхронного програмування для відгуків користувацького інтерфейсу, інтерфейвів різної складності за допомогою власних контролів і сторонніх компонентів, використання WPF для XAML і Silverlight додатків, створення триланкових систем, що працюють зі стандартними веб-сервісами за допомогою WCF-серверів та додатків власної розробки, використання компонентів з відкритим похідним кодом та керування залежностями компонентів за

допомогою nuget, таких як EntityFramework, Prism, BLToolkit, NLog, Autofac, FluentValidation, Automapper, CaliburnMicro, власно розроблені автооновлення додатків, створення пакетів для інсталяції MSI за допомогою WiX.

Для розробки можна використати платформу .NET, PHP або Mono.

Платформа .NET Framework - це технологія для підтримки створення і виконання додатків нового покоління і веб-служб XML. При розробці платформи .NET Framework враховувалися наступні цілі:

- Забезпечення узгодженої об'єктно-орієнтованого середовища програмування для локального збереження і виконання об'єктного коду, для локального виконання коду, розподіленого в Інтернеті, або для віддаленого виконання;
- Забезпечення середовища виконання коду, що мінімізує конфлікти при розгортанні програмного забезпечення та управлінні версіями;
- Забезпечення середовища виконання коду, що гарантує безпечне виконання коду, включаючи код, створений невідомим або не повністю довіреною стороннього постачальника;
- Забезпечення середовища виконання коду, що виключає проблеми з продуктивністю середовищ виконання сценаріїв або інтерпретується коду;
- Забезпечення єдиних принципів розробки для різних типів додатків, таких як додатки Windows і веб-додатки;
- Взаємодія на основі промислових стандартів, яке гарантує інтеграцію коду платформи .NET Framework з будь-яким іншим кодом.

Mono включає в себе компілятор мови C# - dmcs, середовище виконання .NET - mono (з підтримкою JIT) і mint (без підтримки JIT), відладчик, а також ряд бібліотек, включаючи реалізацію WinForms, ADO.NET і ASP.NET, а також компілятори smcs (для створення додатків для Moonlight) і vbc (для додатків, написаних на VB.NET).

В рамках проекту також розробляються прив'язки для графічної бібліотеки GTK + на платформу .NET.

Також Mono містить альтернативу структурі WPF-додатків (XAML + C # або будь-який інший мову, який підтримується в даному середовищі виконання). Дана мова називається Glade, за допомогою нього можна збирати GTK-додатки.

Mono може виконувати модулі, написані на мовах C #, F #, Visual Basic .NET, Java, Boo, Nemerle, Python, JavaScript, Erlang, SmallTalk, Lisp, PHP і Object Pascal (при наявності компілятора в середовищі .Net / Mono). Очікується також підтримка мов C, Ada 2005 і Eiffel.[8]

Наступним важливим, можна сказати – основним, етапом розробки веб-сайту буде розробка програмного забезпечення сайту. Для початку, оберемо зручне середовище розробки. Також проведемо варіантний аналіз програмних засобів реалізації та систем керування базами даних.

Обравши найбільш підходящий варіант для реалізації поставлених завдань, реалізуємо основні модулі веб-сайту.

Node.js - це платформа для створення швидких та масштабованих серверних додатків за допомогою JavaScript. Node.js - це час виконання, а npm - менеджер пакетів для модулів Node.js.

Visual Studio Code підтримує нестандартні мови JavaScript і TypeScript, а також налагодження Node.js. Однак, щоб запустити програму Node.js, вам потрібно буде встановити час виконання Node.js на вашому комп'ютері.

Щоб розпочати роботу з цим покроковим керівництвом, встановіть Node.js для своєї платформи. Менеджер пакетів вузлів включений до розподілу Node.js. Вам потрібно буде відкрити новий термінал (командний рядок), щоб інструменти командного рядка node та npm знаходились у вашому PATH.

Щоб перевірити, чи правильно встановлено на вашому комп'ютері Node.js, відкрийте новий термінал і введіть node--version, і ви побачите встановлену поточну версію Node.js.

PHP має такі особливі властивості:

- Простий і потужний інструмент. Годиться як для вирішення новачком примітивних завдань, так і для розробки складних проектів. Добре читається синтаксис;
- Швидко розвивається. Додаються всі нові функції, команда розробників оперативно реагує на bug-report'и;
- Має досить стерпне (для скриптового мови) об'єктну модель (вже трохи краще, ніж JavaScript;)), тоді як без ООП зараз майже нікуди;
- Стійкий і надійний при вирішенні стандартних завдань - як наслідок великої кількості користувачів і оперативності розробників;
- Підтримується переважною більшістю провайдерів хостингу;
- Дуже зручна робота з БД, особливо MySQL.

Javascript популярний не випадково, а завдяки своїм безперечним перевагам.

- Незамінність для веб-розробки. Підтримка скриптів усіма популярними браузерами; повна інтеграція з версткою сторінок (HTML + CSS) і серверної частиною (backend);
- Швидкість роботи і продуктивність. Javascript дозволяє частково обробляти веб-сторінки на комп'ютерах користувача без запитів до сервера. Це економить час і трафік, знижує навантаження на сервер;
- Потужна інфраструктура (екосистема). Перші 10 років про це навіть не згадувалось. Потім кількість готових рішень в відкритому доступі так зросла, що працювати з Javascript і його фреймворками стало задоволенням;
- Простота і раціональність застосування. Просту завдання можна вирішити за 5 хвилин, не треба робити зайву роботу. Для складних завдань є варіанти вирішення, можна підібрати кращий, адаптувати;

—Зручність для користувача інтерфейсів. Заповнення форм, вибір дій, активація кнопок, перевірки введення, реагування на наведення / кліки миші та інше. Це дає приголомшливий рівень юзабіліті;

—Легкість освоєння. Можливо, це справа звички. Спочатку код може здатися складним, але до його синтаксису, логіки швидко звикаєш. Відображення дій відчутно додає ентузіазму;

Як будь-яка мова програмування, Javascript має деякі недоліки (обмеження):

—Немає можливості читання та завантаження файлів. Це обмеження функціональності на стороні клієнта. Головна причина - міркування безпеки;

—Нестрога типізація і вільне трактування. Мова ігнорує явні невідповідності. Має місце різна інтерпретація даних. Немає можливості раннього виявлення помилок. Всі недоліки виявляються вже на етапі роботи;

—Ні підтримки віддаленого доступу. Тому, мова не можна використовувати для мережеских додатків. За це Javascript навіть не вважають повноцінною мовою програмування;

—Доступність для зловмисників. Вільна скриптова мова найпростіше вбудувати фрагмент шкідливого коду, який може нашкодити користувачеві. Надія тільки на антивірус і фаєрвол.[4]

Треба відзначити, що підприємницькі кола користувачів Javascript активно покращує мову, усуваючи багато недоліків. Вузких місць стає все менше. Браузери постійно вдосконалюють роботу з JS. Вихід HTML[5] 5 дав новий поштовх до розширення можливостей скриптів.

При виборі професії web-програміста без Javascript не обійтися. Знадобляться також знання з HTML і CSS, ці інструменти зазвичай працюють в зв'язці. Швидше засвоювати мову сприяють:

- простоту синтаксису і раціональність, інтуїтивна зрозумілість коду;
- простота і доступність інструментів (мінімум: текстовий редактор і браузер);
- наявність в інтернеті великої кількості матеріалів для навчання і практики.

Для розробки проєкту було обрана мова JavaScript.

1.4 Постановка задачі та вимоги до розробки інформаційної системи

Задачею кваліфікаційної роботи бакалавра є прорахунок собівартості продукту, на платформі JavaScript, та виконує групи функцій:

Робота з підрахунку вартості продукту на основі вибраного функціоналу та параметрів.

До групи функцій роботи з підрахунку вартості продукту на основі вибраного функціоналу та параметрів належать:

- вибір сфери застосування;
- вибір платформи;
- вибір та компонування функціональності майбутнього продукту;
- вибір якості інтерфейсу;
- детальний опис у файлі про вибір функцій;
- отримання файлу оцінки.

Крім того, робочі функції можуть забезпечувати розрахунок вартості функцій, які враховуються у виробі.

Детальні специфікації взаємодії товару з обраними функціональними групами визначені в документах специфікації функціональної групи товару.

Функціональні групи можна використовувати для зменшення різниці в часі між реалізацією продукту та прогнозованим використанням.

Наприклад, функціональні групи мають більш стабільну стабільність, оскільки функціональна група продовжує розвиватися доти, поки не буде обрано платформу та функціональні групи.

Специфікації функціональної групи використовуються при аналізі вимог для вибору платформи та функціональних груп товару, у технічній документації на виріб та в інших документаціях.

Групова конфігурація товару, що реалізується, є лише однією з функціональних груп, інші - це специфікація вимог ринку / споживача, промислові стандарти та функціональні групи партнерів.

Отже, залежно від вимог програми до продукту, функціональні групи та функції будуть відрізнятися залежно від використання продукту користувачами.

Для проєктування роботи було використано: JavaScript, React[6], SCSS[7], Visual Studio Code, HTML.

Розділ 2

Проектування інформаційної системи

2.1 Функціональна структура та бізнес-процеси системи

Для розрахунку собівартості продукту потрібні наступні бізнес-процеси:

- вибір сфери застосування;
- вибір платформи застосування;
- вибір функціональних можливостей;
- вибір необхідних складових для реалізації кожної функцій;
- вибір якості дизайну;
- вивід короткої інформації по створення продукту;
- отримання детального опису у PDF форматі.

Бізнес-процес «Вибір сфери застосування». Даний бізнес-процес надає можливість обрати сферу застосування майбутнього продукту з вже запропонованих або обрати свій абсолютно індивідуальний проект.

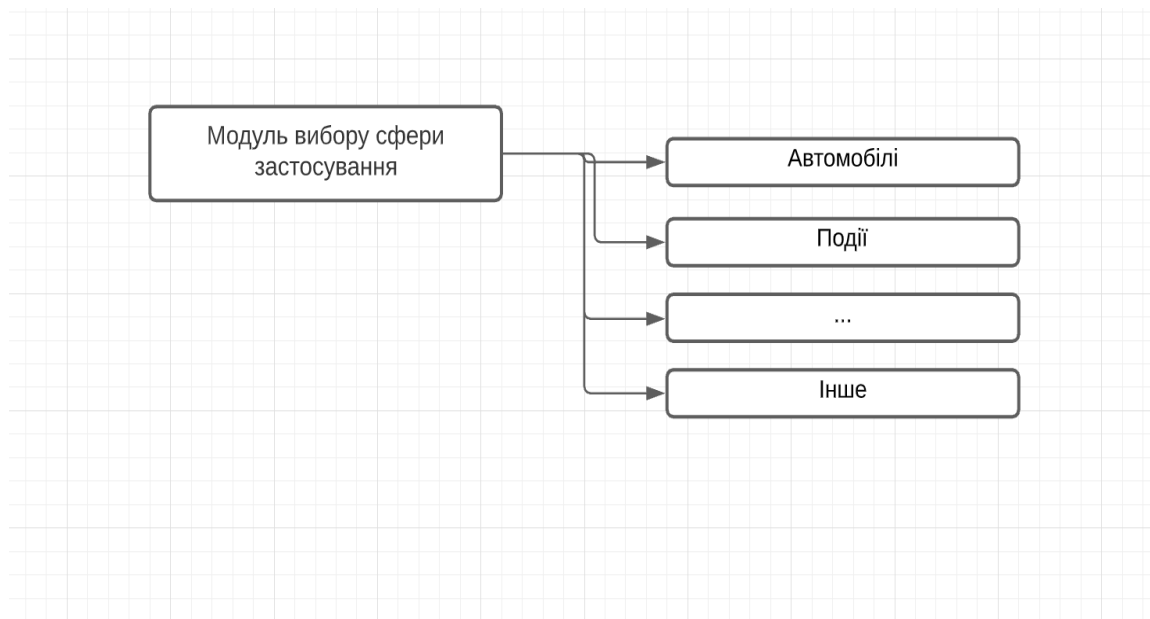


Рисунок 2.1.1 — Модуль вибору сфери застосування

Бізнес-процес «Вибір платформи застосування». Даний бізнес-процес надає можливість обрати операційні системи які будуть підтримувати застосунок.

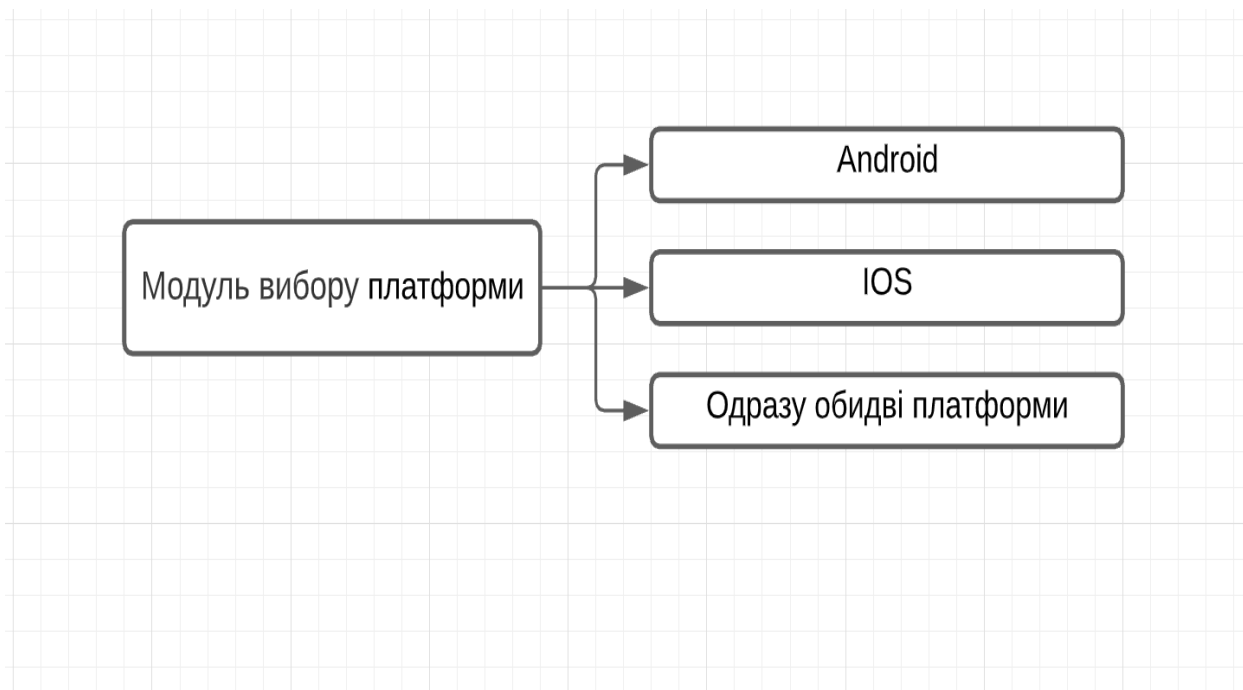


Рисунок 2.1.2 — Модуль вибору платформи

Бізнес-процес «Вибір функціональних можливостей». Найбільш широкий бізнес-процес який дає можливість обрати функціональність майбутнього продукту.

На цьому етапі формується структура майбутнього продукту та від нього залежать можливі подальші зміни в бізнес-процесі «Вибір необхідних складових для реалізації кожної функцій».

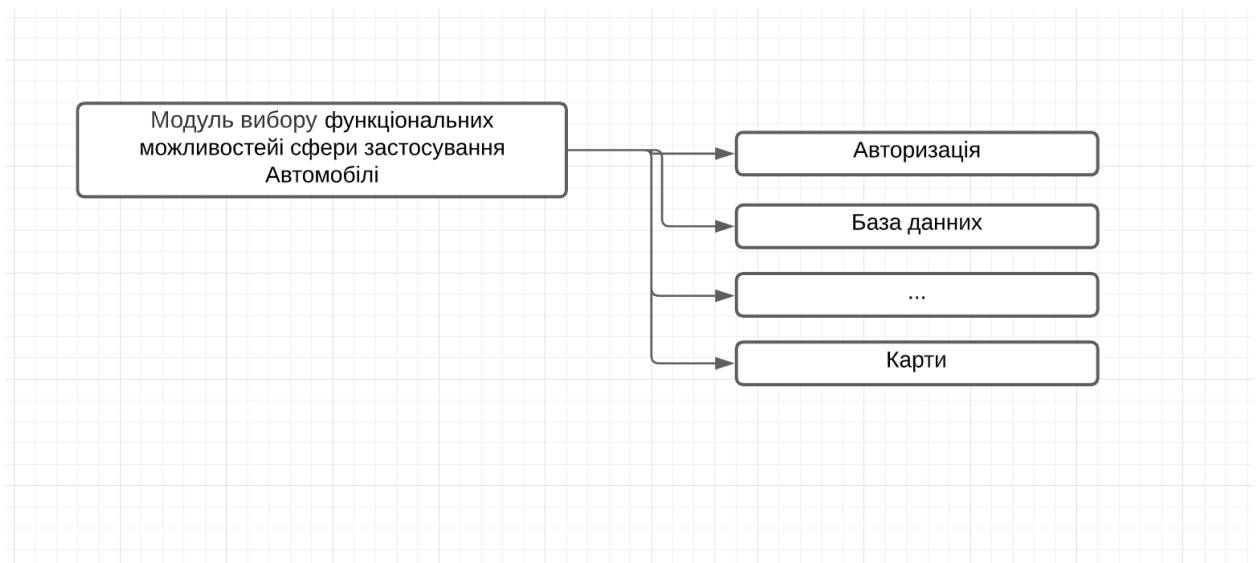


Рисунок 2.1.3 — Модуль вибору функціональних можливостей

Бізнес-процес «Вибір необхідних складових для реалізації кожної функції» відповідає за надання набору мінімальних функцій застосунку в залежності від обраної категорії та можливістю додавання функціоналу застосунку.

Функціональні особливості можна комбінувати та застосовувати відповідно до функціональних вимог у кожному шарі.

Цей процес називається вибором компонентів. Від вибору компонентів залежить наповнення структури вибраних функціональних можливостей додатковими можливостями, як показано в блок-схемі на рисунку 2.1.4.

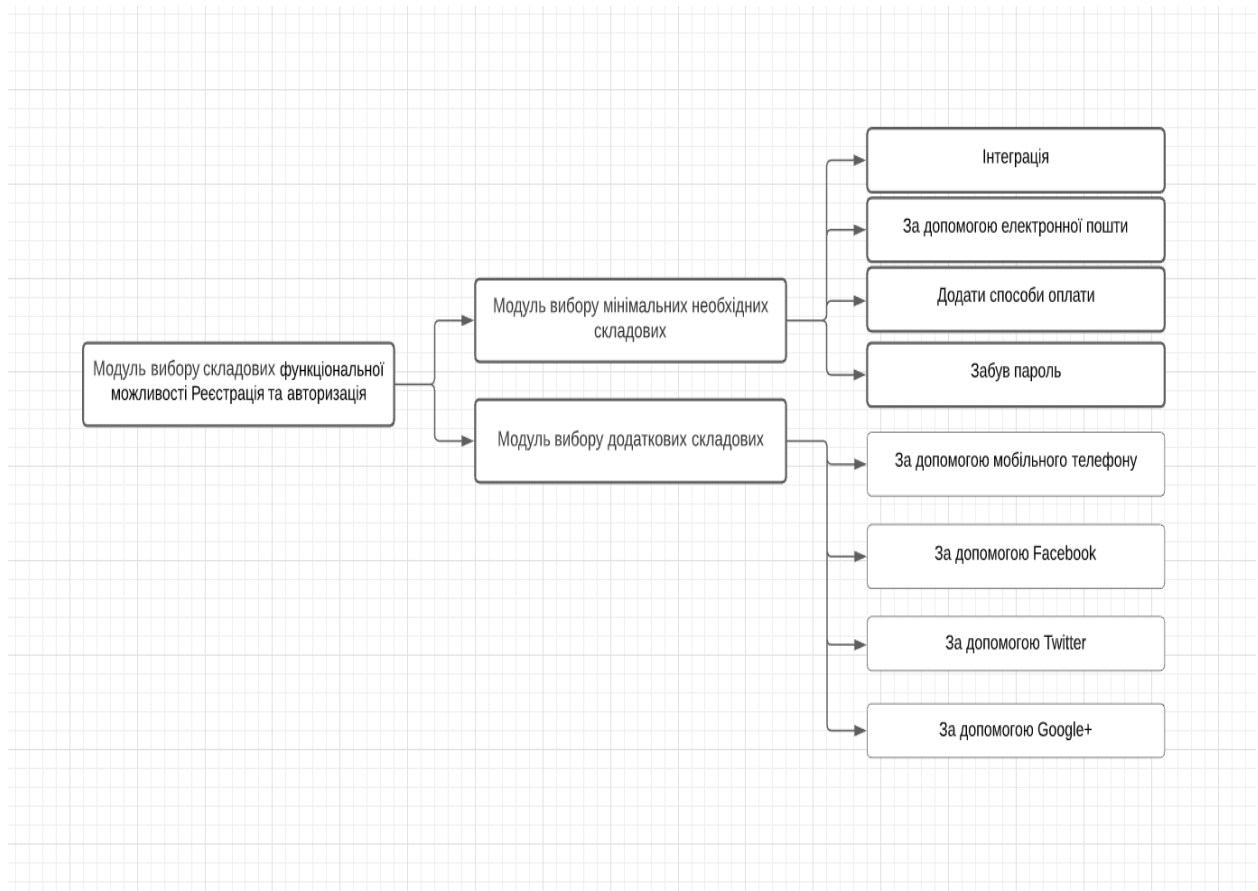


Рисунок 2.1.4 — Модуль вибору складових

Бізнес-процес «Вибір якості дизайну» пропонує замовнику обрати вигідний для нього дизайн майбутнього продукту, що відповідно впливає на ціну проекту. Дизайн може бути складним, простим, або клієнт може взагалі відмовитися від його розробки.

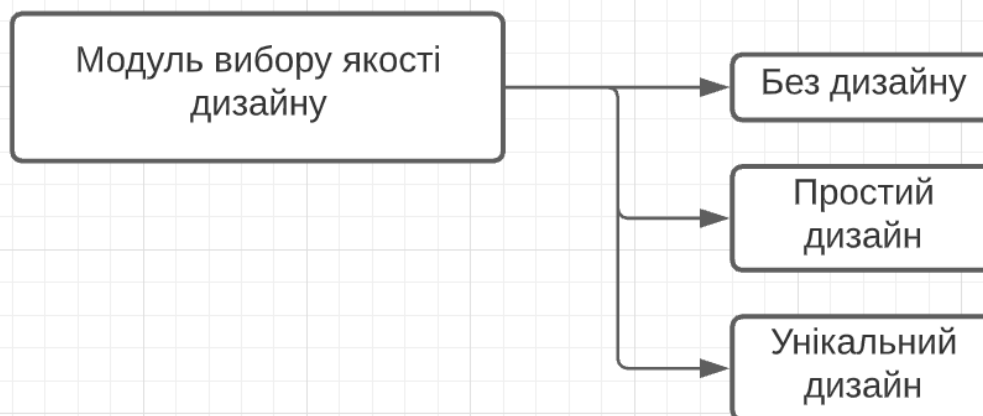


Рисунок 2.1.5 — Модуль вибору якості дизайну

Бізнес-процес «Вивід короткої інформації по створення продукту» Надає коротку інформацію по ціні продукту які включають у себе основні показники, а саме:

- час розробки продукту;
- ціна реалізації продукту;
- спеціалісти, що необхідно долучити до проекту;
- найближча можлива дата початку роботи.

Бізнес-процес «Отримання детального опису у PDF форматі» надає повну інформацію замовнику про обрані функції у його замовленні. Кількість годин які буде виділятися на кожен процес конструювання застосунку.

Нижче зображено блок схеми модулів системи.

Модуль вибору сфери застосування відображає роботу даного модуля, як показано на рисунку 2.1.6 він відповідає за роботу вибору сфери застосування.

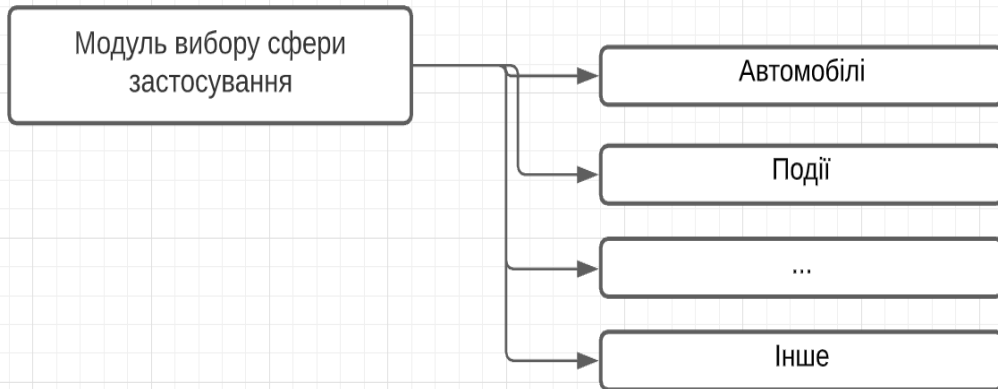


Рисунок 2.1.6 — Модуль вибору сфери застосування

Модуль вибору платформи відображає роботу даного модуля, як показано на рисунку 2.1.7 він відповідає за роботу вибору платформи.

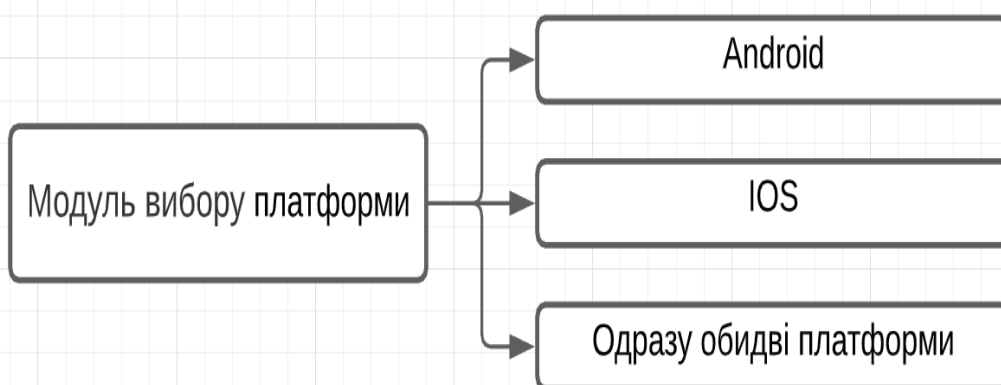


Рисунок 2.1.7 — Модуль вибору платформи

Модуль вибору функціональних можливостей відображає роботу даного модуля, як показано на рисунку 2.1.8 він відповідає за роботу вибору функціональних можливостей, кожна з яких складається зі складових, як це зображено на рисунку 2.1.9.

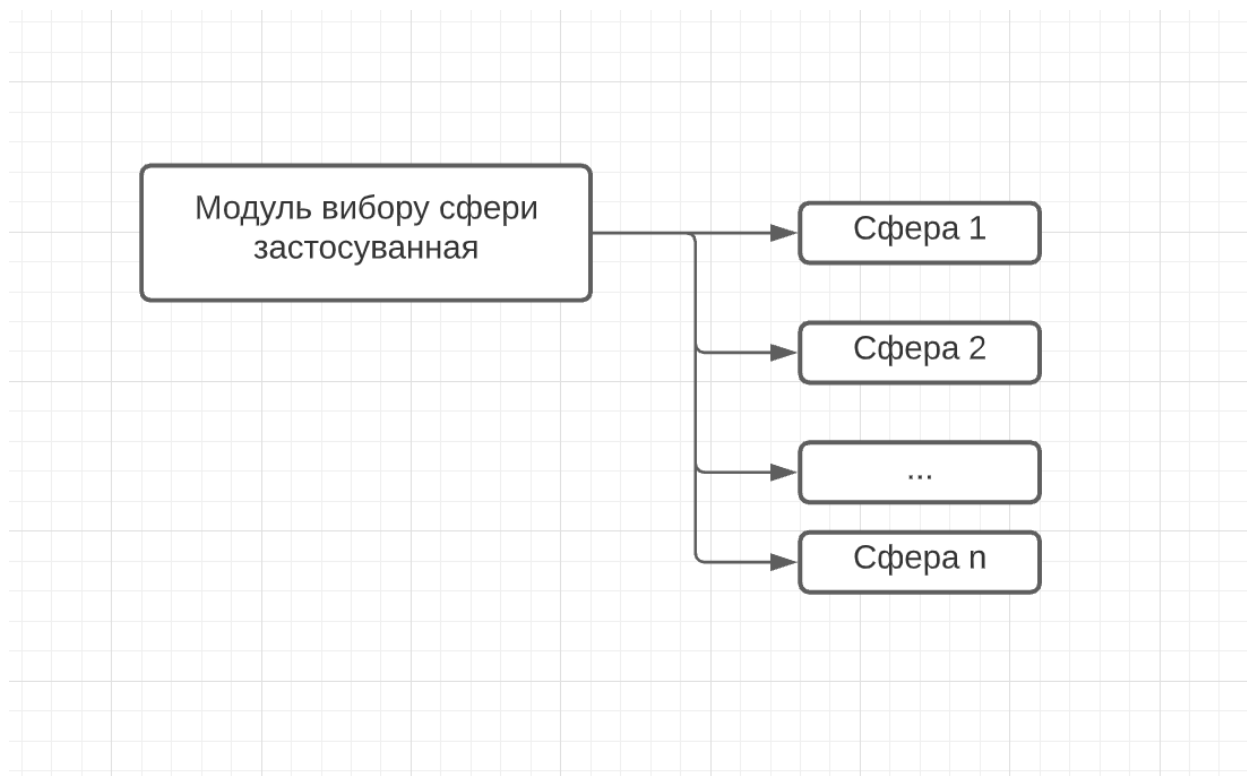


Рисунок 2.1.8 — Модуль вибору сфери застосування

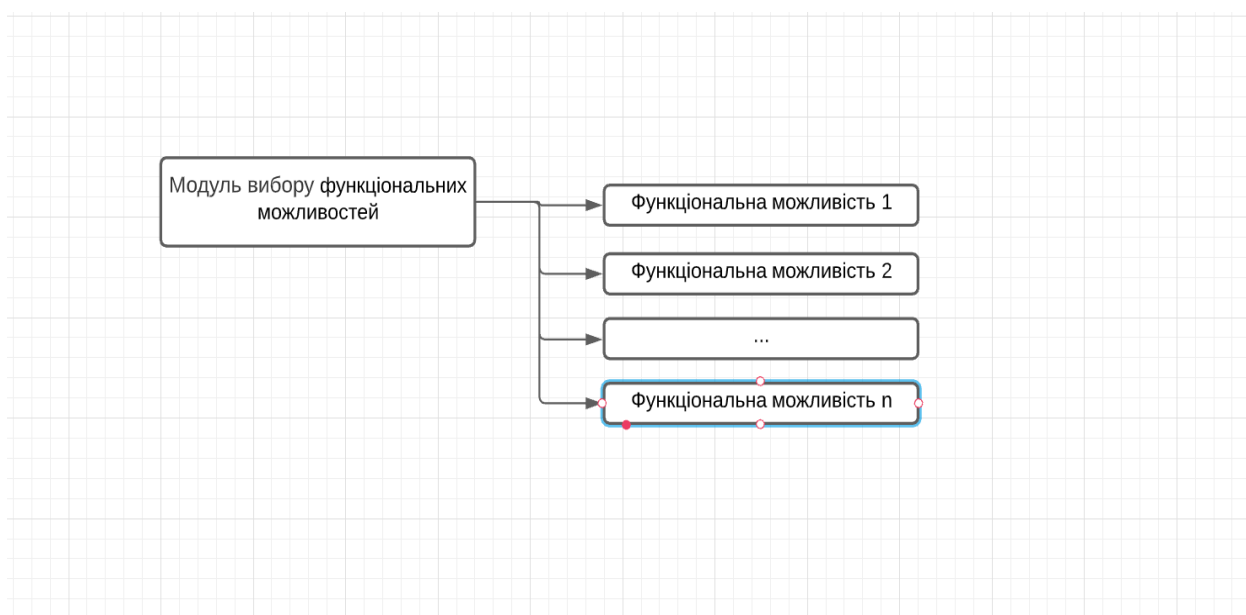


Рисунок 2.1.9 — Модуль вибору функціональних можливостей

Модуль вибору якості дизайну відображає роботу даного модуля, як показано на рисунку 2.1.10 він відповідає за роботу вибору якості дизайну.

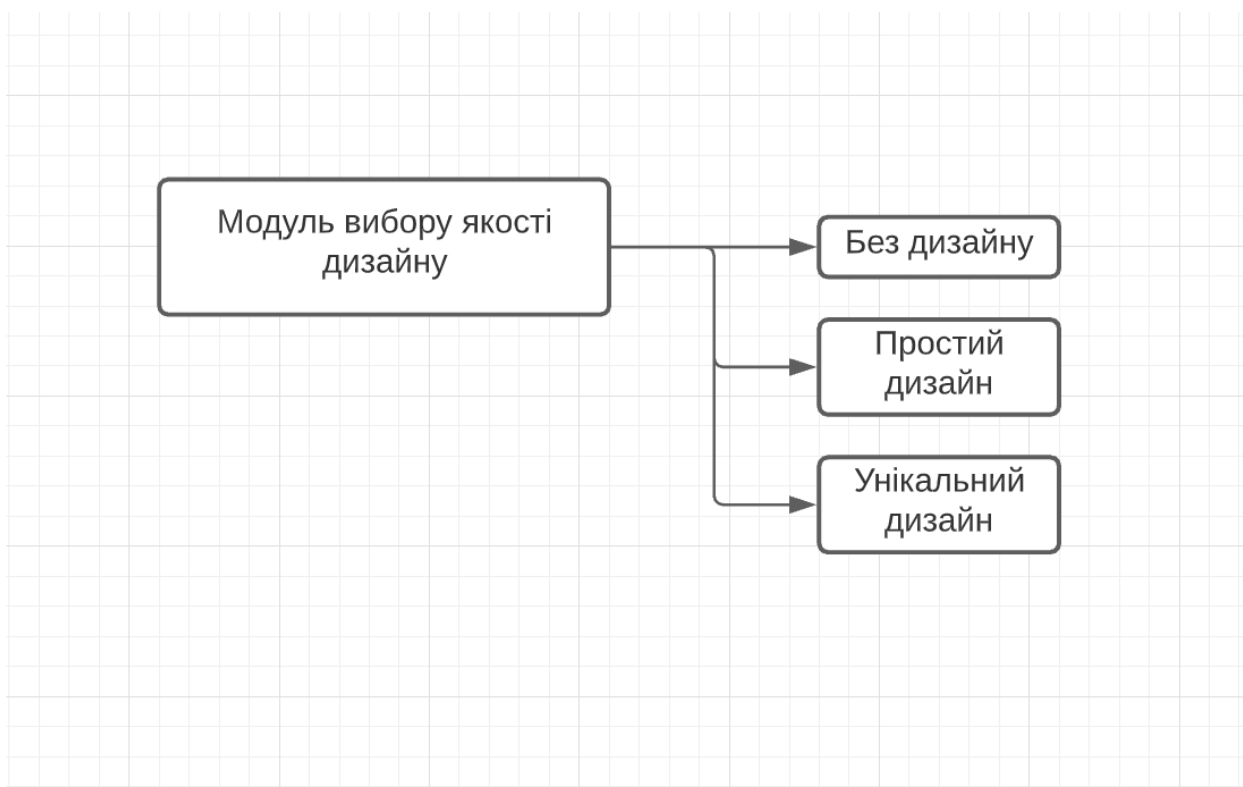


Рисунок 2.1.10 — Модуль вибору якості дизайну

2.2 Інформаційна структура системи

Головна сторінка спілкування із замовником (рисунку 2.2.1, рисунку 2.2.3) дає вам можливість вибрати одну з готових сфер, де використовуються додатки. Якщо ви виберете готову сферу, система автоматично керуватиме різними фазами розвитку. Якщо ви маєте на увазі конкретні запити, це можна зробити на вкладці «Інше». На сторінці також відображається час, необхідний для заповнення заявки. Ви можете працювати на головній сторінці у вільний час або коли можете вільно переходити до інших завдань.

У кожній індустрії прописаний вже мінімальний функціонал та можливість додавання функцій у застосунку.

При виборі одної з індустрій – з'являється вікно з обов'язковим вибором платформи на якій буде використовуватися продукт.

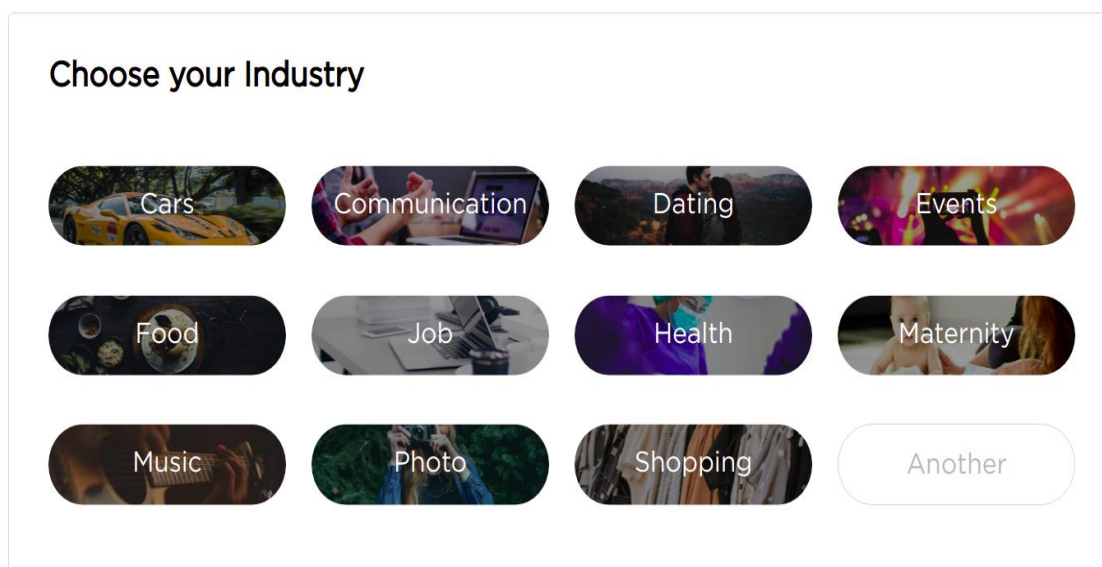


Рисунок 2.2.1 — Екран вибору сфери

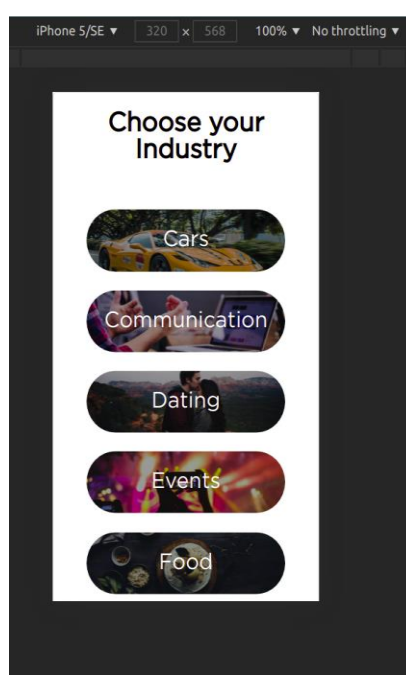


Рисунок 2.2.2 — Мобільна версія екрану вибору сфери

Вибір платформи складається з 3-ох запропонованих варіантів (рисунок 2.2.3, рисунок 2.2.4):

—Android;

—IOS;

—одразу обидві платформи.

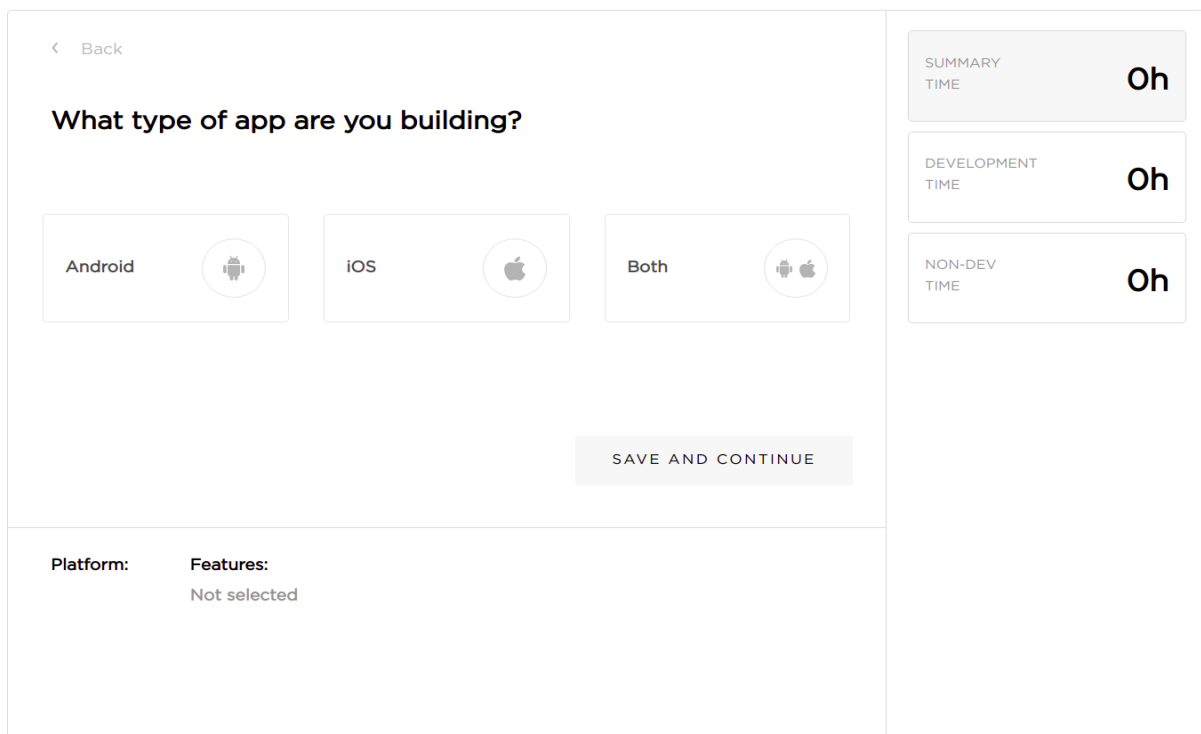


Рисунок 2.2.3 — Мобільна версія екрану платформи

The screenshot displays a configuration interface for a software product. At the top left, there is a 'Back' button. The current step is labeled 'STEP 2 OFF 9'. The main question is 'Will you send Notifications to your users?'. Below this question are two buttons: 'Yes +43h' and 'No'. On the right side, there is a summary panel with three rows: 'SUMMARY TIME 91.2h', 'DEVELOPMENT TIME 57.2h', and 'NON-DEV TIME 34h'. At the bottom, there are two sections: 'Platform:' with 'Android' selected, and 'Features:' with 'Registration & Authorization' selected and marked with an 'X'.

Category	Value
SUMMARY TIME	91.2h
DEVELOPMENT TIME	57.2h
NON-DEV TIME	34h

Platform: Android

Features: X Registration & Authorization

Рисунок 2.2.4 — Экран выбора платформы

Після вибору сфери та платформи програмного продукту

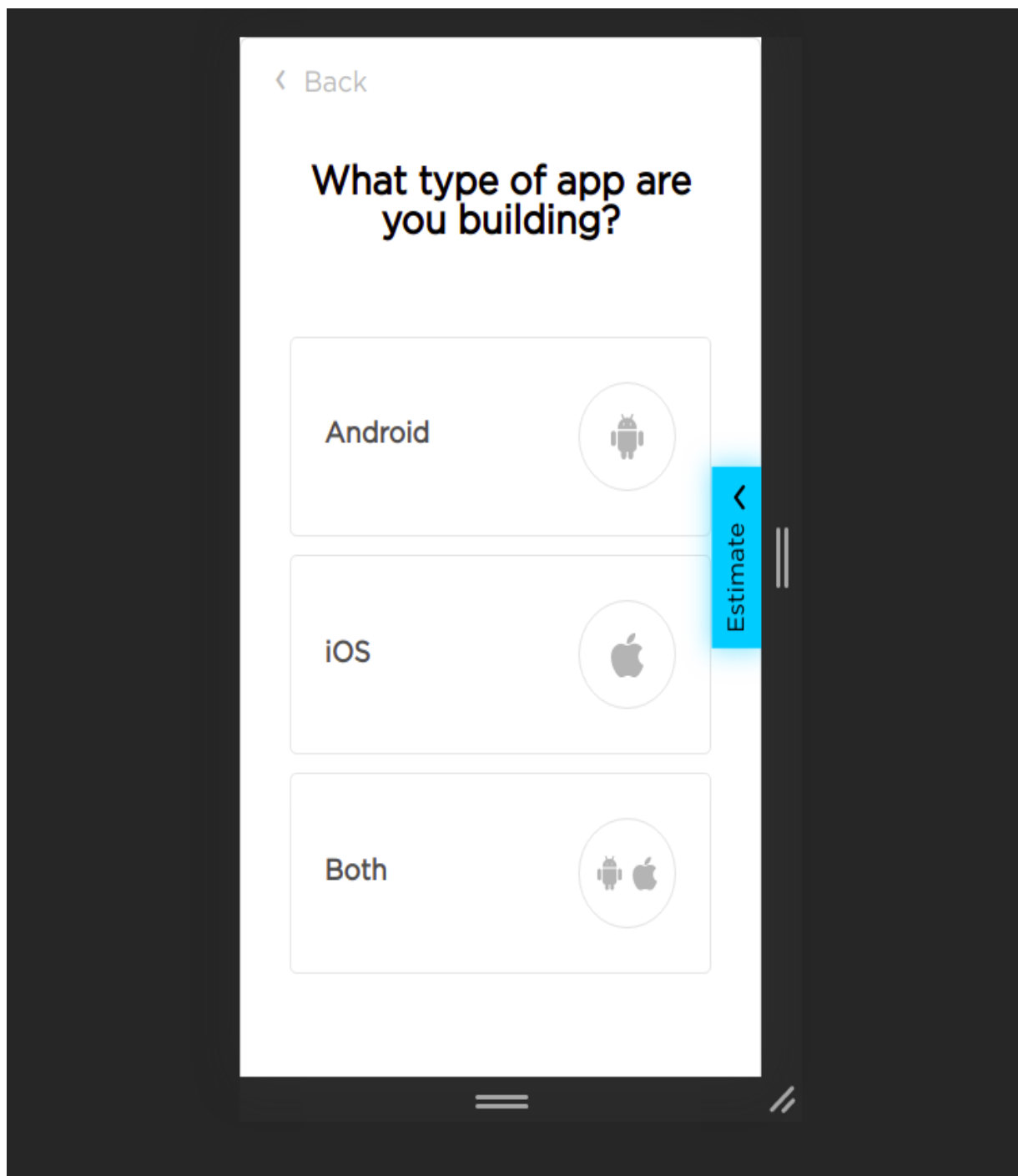


Рисунок 2.2.5 — Экран выбора функциональных возможностей

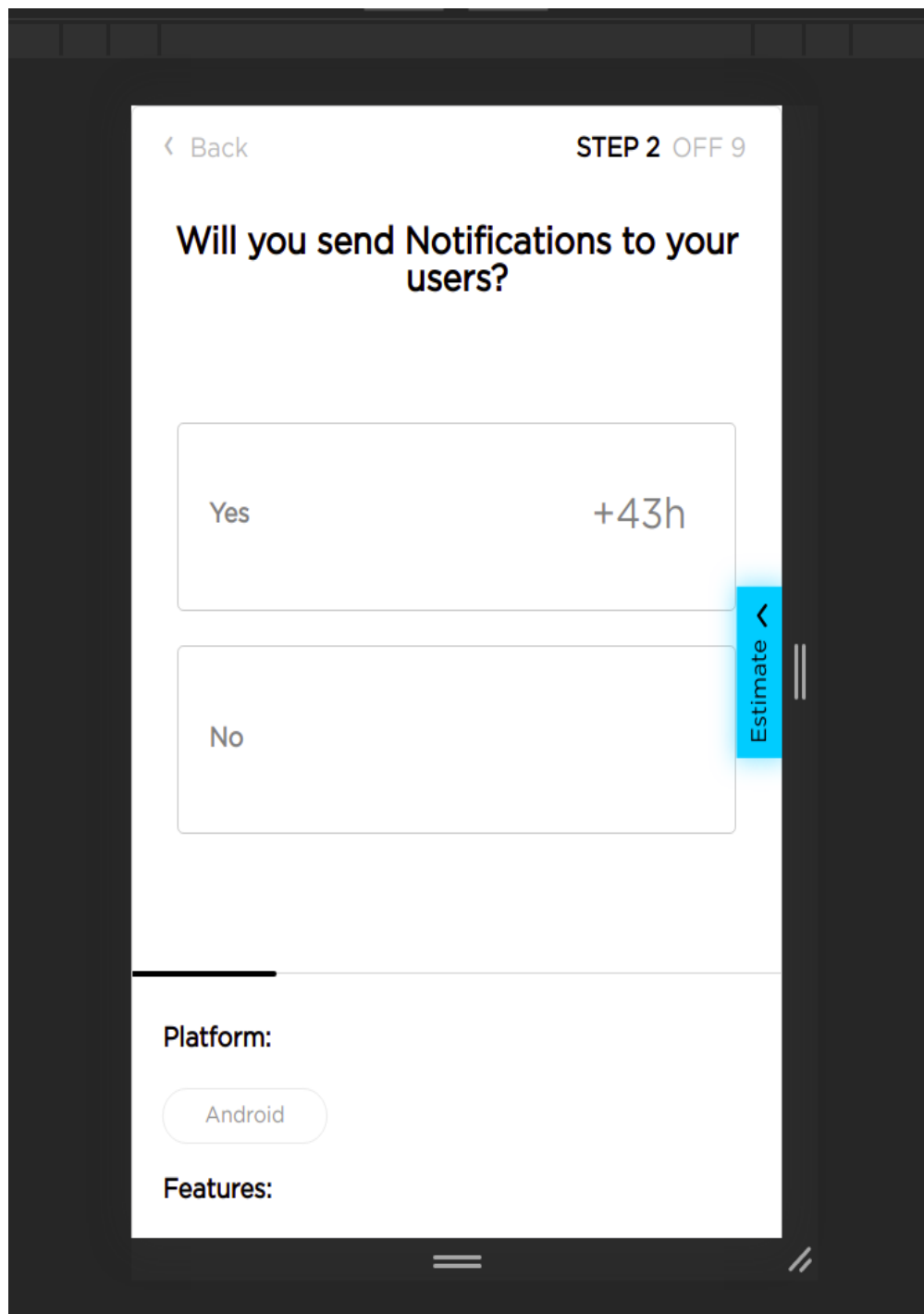


Рисунок 2.2.6 — Мобільна версія екрану вибору функціональних можливостей

Далі в залежності від вибору індустрії дається можливість вдосконалити продукт.

Обов'язковим вибором є дизайн застосунку:

—без дизайну;

- з простим дизайном;
- індивідуальний дизайн.

Do you need a design for your App?

None +0 Simple +8 Custom +19

GET ESTIMATE

SUMMARY TIME	116.6h
DEVELOPMENT TIME	78.2h
NON-DEV TIME	38.4h

Platform: ios Features: X Registration & Authorization

Рисунок 2.2.7— Екран вибору якості дизайну

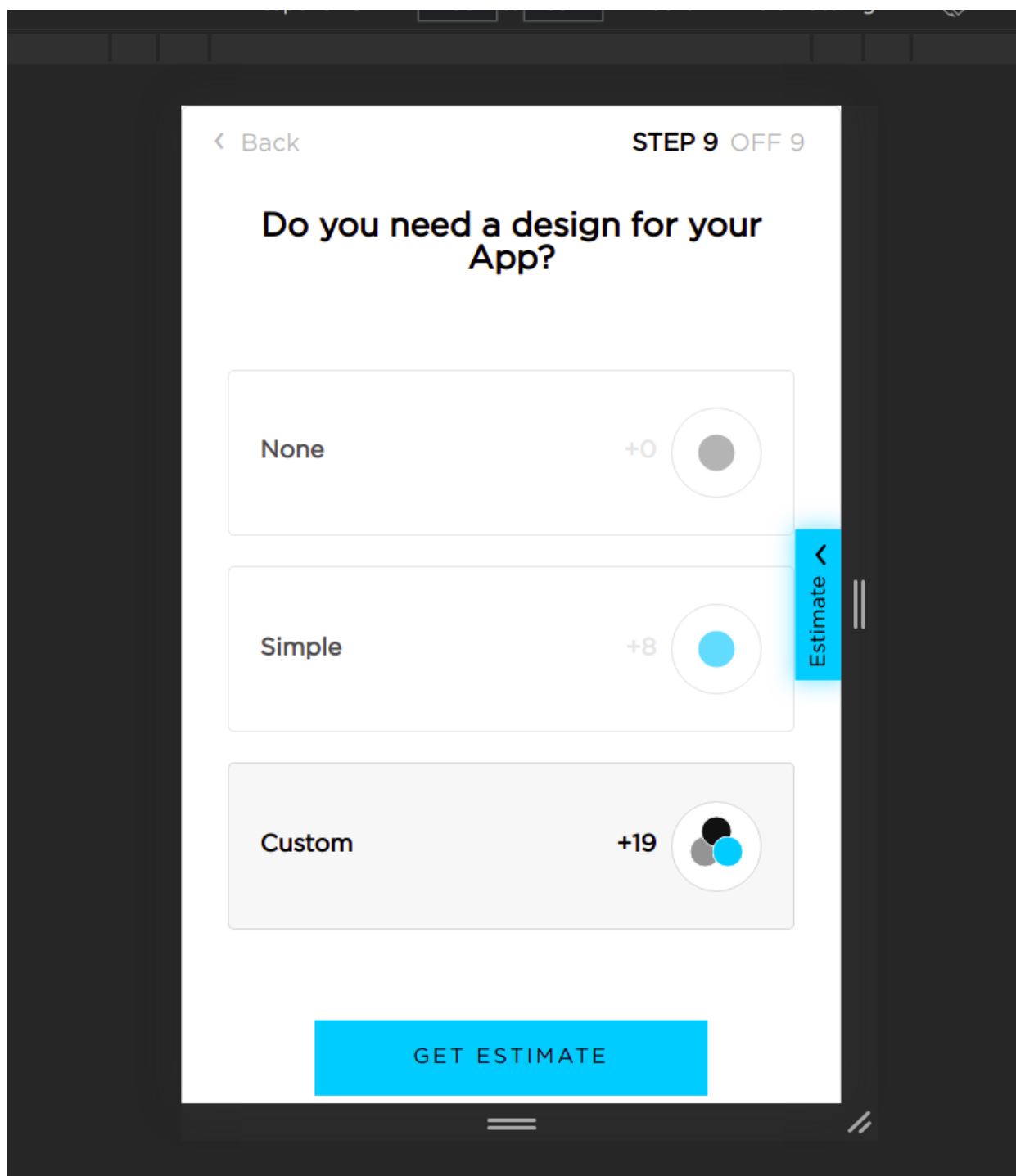



Рисунок 2.2.8 — Мобільна версія екрану вибору якості дизайну

В кінці шляху розрахунку вартості товару через вікно з короткою базовою інформацією про ціну, робочий час та кількість спеціальних спеціалістів у проектах, а також більш детальну роздрібну ціну товару. Прозорість основних характеристик товарів допомагає споживачам краще оцінити ціну товару. Чим більше прозорості, тим краще.

← Back

Rough Estimate of My App

SUMMARY TIME 116.6h	OPTIMISTIC 104.9h	YOUR DEDICATED TEAM Android developer ✓ ios developer ✓ Web developer ✓ Project Manager ✓ QA Engineer ✓ UI/UX Designer	 Leave your email to download your Rough Estimate with full Features list
	PESSIMISTIC 128.3h		

Estimated release date 19 May 2021. Leave request to start a project in 2 days.

DEVELOPMENT PRICE \$ 782	MINIMUM \$ 668
	MAXIMUM \$ 896

Your Email

GET MY ESTIMATE

Рисунок 2.2.9 — Екран відображення результатів

Товари з чітким і повним описом їх характеристик, часу, необхідного на переробку, та обсягу виробництва та просування, і нарешті, причини виробництва цього товару. Цей метод супроводжується використанням алгоритму визначення їх характеристик.

Here is a rough estimate of your project!
 Thank you for finding out the cost ! We got the result and
 We think that we can make an ideal estimate or even a
 discount.

Your chosen category **Cars**
 for which you have chosen a **custom** design
 and **ios** platform

Registration & Authorization	
Integration	6.5h
via Email	35.0h
Add payment details	19.0h
Forgot password	9.9h
Non development time	
Project setup	8.3h
Architecture	5.1h
Database	5.3h
Swagger	4.3h
Network	1.8h
Meetings	4.8h
Continuous integration	3.6h
Continuous delivery	1.8h
Release per sprint	6.7h
Release per integration	1.8h

Backend:	\$285
Frontend:	\$50
Design:	\$190
Mobile:	\$179
Management:	\$78
Total:	\$782USD

If you have any questions, reply to this email or contact us at mail@mail.com.

© Copyright. All rights reserved.

Рисунок 2.2.10 — Файл відображення результатів

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

При виборі мови програмування було обрано JavaScript.

JavaScript - це мова програмування, яка дозволяє створювати веб-історію в Інтернеті, оскільки вона відповідає на дії користувача.

Програми JavaScript невеликі, легкі та швидкі, оскільки не мають жодних циклів та інших повільних частин.

Вони також ефективніші, оскільки відпадає необхідність в інтерпретаторах, щоб запустити їх, як Python або Ruby. JavaScript дійсно хороший для таких речей, як інтерактивний дизайн веб-сторінок, і для написання автоматизованих тестів.

Послідовність інструкцій (які називають скриптом, програмою або сценарієм), що виконує інтерпретатор, який вбудований в звичайний Веб-браузер. Іншими словами, код що виконується, вбудовується в документ HTML і виконується на стороні клієнта.

Для виконання програми не потрібно перезавантажувати Веб -сторінку тому що, всі програми починають свою роботу в відповідь на будь-яку подію.

Наприклад, перед відправленням даних форми є можливість перевірити їх на вілідність введених даних, якщо значення задовільняють очікування, заборонити їхнє відправлення.

JavaScript - це скриптова мова програмування і діалектом цієї мови є ECMAScript.

JavaScript зазвичай використовується для програмного доступу до об'єктів додатків як вбудовану мову. Найбільш широким застосуванням у браузерах знаходить як мову сценаріїв для забезпечення інтерактивності веб-сторінок.

Основні архітектурні риси:

- автоматичне керування пам'яттю;
- прототипне програмування;

—функції як об'єкти першого класу.

На розвитку JavaScript відобразився вплив багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання не кваліфікованими спеціалістами в цій області.

JavaScript має низку властивостей, які притаманні ООП мовам, але реалізація в мові прототипування стає причиною відмінності в роботі з об'єктами в порівнянні з звичайними ООП мовами.

Крім того, JavaScript має ряд властивостей, що притаманні функціональним мовам програмування, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання - що додає мові додаткову гнучкість. JavaScript має ряд повноважень об'єктно-орієнтованої мови, але усвідомлює мовою взуття прототипів важливість роботи з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами.

Крім того, JavaScript має низку повноважень, потужну функціональність - функціонує як першокласні об'єкти, об'єкти як списки, перенесення, анонімні функції, заміна - що додає свої додаткові недоліки. Основною парадигмою об'єктно-орієнтованих мов є моделювання ієрархії класів. Батьківський клас представляє роль батьківського об'єкта об'єкта. Дочірні класи мають виконувати обов'язки, які не пов'язані безпосередньо з батьківським об'єктом.

Вони, як правило, більш абстрактні, і аспект їхньої діяльності не такий важливий.

Вони оптимізовані та перетворені на бібліотеки, які не залежать від батьківського класу.

Головне - це те, що впровадження цих бібліотек є окремим від дочірних класів.

Мінуси JavaScript:

—модульна система - JavaScript не надає можливості керувати залежностями та ізолювати зони видимості;

- стандартна бібліотека - зокрема, відсутній інтерфейс для програмування додатків для роботи з файловою системою, управління потоком вводу-виводу, основні типи двійкових даних;
- стандартні інтерфейси до веб-серверів та баз даних;
- система управління пакетами, яка буде відстежувати залежності та автоматично встановлювати їх.

Найважливішою системою управління пакетами JavaScript для веб-розробників є npm. npm пропонує можливість управління великою кількістю проектів і дуже простий у використанні.

Оновлення: деякі користувачі писали, що npm не є правильним рішенням. npm дуже простий у використанні, але в довгостроковій перспективі це не правильне рішення. Ви не можете ефективно керувати великим проектом за допомогою npm.

npm - це менеджер пакетів, який може керувати вашими залежностями та їх установкою.

Синтаксис мови JavaScript дуже нагадує синтаксис Сі і Java, семантично ж мова набагато ближче до Self, Smalltalk або навіть Ліспу.

В JavaScript :

- всі ідентифікатори Реєстрозалежні;
- в назвах змінних можна використовувати літери, підкреслення, символ долара, арабські цифри, назви змінних не можуть починатися з цифри.

Структурно JavaScript можна представити у вигляді об'єднання трьох частин, що чітко різняться одна від одної:

- ядро (ECMAScript),
- об'єктна модель браузера (Browser Object Model або BOM),

- об'єктна модель документа (Document Object Model або DOM).

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа.

2.3.2 Вибір бібліотеки

React (старі назви: React.js, ReactJS) — це бібліотека JavaScript, яка використовується для створення користувацьких інтерфейсів, та створена для вирішення проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників.[7]

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках.

Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків. React дозволяє розробникам створювати великі веб-додатки, які використовують дані, що змінюються з часом, без перегляду сторінки.

Його призначення не може бути швидким, простим, масштабним. React обробляє лише користувальницький інтерфейс у додатках. Ми можемо використовувати відповідні візуальні дані у шаблоні подання моделі контролера (MVC), і їх можна використовувати разом з іншими бібліотеками JavaScript або у великих MVC-фреймворках, таких як AngularJS. Він також може використовуватися

з надбудовами на основі React для з'єднання деталей без вбудованого інтерфейсу веб-програми.

Однак правильний інструмент - це розумний інструмент, який допоможе вам задовольнити потреби вашого додатка та вимоги цільової аудиторії. Для досягнення успіху не завжди потрібно використовувати найновіший фреймворк або доповнення до фреймворку. Правильно підібраний інструмент служить вам добре та забезпечує покращення в 10 разів. React може забезпечити вражаюче підвищення продуктивності дуже легкого веб-дodatку. У багатьох випадках React забезпечує кращу продуктивність, ніж Angular або Ember. Однак, якщо ваша команда не зацікавлена в навчанні та експериментах з новими технологіями, а скоріше дотримуватиметься знайомого, React може стати правильним вибором для вашого проекту. Це дуже гнучка структура, яка дозволяє розширити можливості.

React не дозволяє вам робити те, що роблять AngularJS та Ember. Це не величезні обмеження, і ви без сумніву можете створювати масштабовані програми. Однак одне обмеження може бути у формі прив'язки даних. React не дає можливості створити синтаксис прив'язки даних. Ці обмеження можна пом'якшити за допомогою обраних вами бібліотек.

React також обмежує кількість рендерингу на стороні сервера, який він підтримує, і часто рекомендується використовувати рендеринг на стороні сервера під час створення додатків. Це викликає занепокоєння, оскільки React не є заміною для Angular або Ember, і він не забезпечує стільки гнучкості при побудові задньої частини програми. Це особливо вірно, якщо ви хочете використовувати рендеринг на стороні сервера в інтерфейсі.

Нарешті, деякі приклади проектів React свідчать про такий недолік: вони можуть бути дорогими, оскільки ви платите за кожен рядок JavaScript. Це може обмежити кількість програм, які може розробляти один розробник. Однак це можуть бути лише виправданнями того, що не вдалося найкраще використати цю розумну технологію. Найбільшим недоліком є відсутність фреймворку JavaScript з великим

співтовариством, що підтримується Google, але React може стати "обов'язковим" фреймворком.

2.3.3 Вибір системи керування базами даних

База даних Firebase Realtime - це база даних, розташована в хмарі. Дані зберігаються у форматі JSON і синхронізуються в реальному часі з кожним підключеним клієнтом. Коли ви створюєте безкоштовні доповнення за допомогою SDK для iOS, Android та JavaScript. За допомогою бази даних у режимі реального часу ви можете створити до трьох унікальних баз даних.

База даних реального часу підтримує OAuth, якщо ваші клієнти вмикають це для автентифікації, налаштувавши форму єдиного входу (SSO) у вашому додатку. Він також має швидку чергу для збереження змін як можна в реальному часі, і включає те, що ми називаємо "чистими" викликами JSON API.

Це означає, що все зберігається у форматі JSON і з ним легко працювати, не потребуючи додаткового проміжного програмного забезпечення OAuth.

Після того, як ваші клієнти налаштують SSO, ви автоматично отримуєте SSO аутентифікацію для всіх їх даних у реальному часі у ваших програмах. Вони також можуть увійти на вашу інформаційну панель та керувати своїми підписками через самообслуговування або мобільний телефон

У реальному часі Замість звичайних HTTP-запитів база даних Firebase Realtime використовує синхронізацію даних - кожен раз, коли дані змінюються, будь підключений пристрій отримує це оновлення протягом мілісекунд. Забезпечте спільну роботу і захоплюючий досвід, не думаючи про мережевому коді.

Не в мережі Додатки Firebase залишаються чуйними навіть в автономному режимі, оскільки SDK Firebase Realtime Database SDK зберігає ваші дані на диску. Після відновлення підключення клієнтський пристрій отримує всі пропущені зміни, синхронізуючи їх з поточним станом сервера.

Доступний з клієнтських пристроїв Доступ до бази даних Firebase Realtime можна отримати безпосередньо з мобільного пристрою або веб-браузера; немає необхідності в сервері додатків. Безпека і перевірка даних доступні через правила безпеки бази даних Firebase в реальному часі, правила на основі виразів, які виконуються при читанні або запису даних.

Масштабування по декількох базах даних З базою даних Firebase Realtime в тарифному плані Blaze ви можете підтримувати потреби вашого застосування в даних в потрібному масштабі, розділяючи дані між кількома екземплярами бази даних в одному проекті Firebase.

Оптимізуйте аутентифікацію за допомогою Firebase Authentication в своєму проекті і аутентифікують користувачів в ваших примірниках бази даних. Керуйте доступом до даних в кожній базі даних за допомогою налаштованих правил Firebase Realtime Database Rules для кожного екземпляра бази даних.

Firebase Realtime це БД, що дозволяє створювати багатофункціональні програми для спільної роботи та забезпечувати безпечний доступ до БД безпосередньо з клієнтського коду.

Дані зберігаються локально, і навіть в автономному режимі події в реальному часі продовжують активуватися, що дає кінцевому користувачу можливість швидко реагувати.

Коли пристрій відновлює з'єднання, база даних реального часу синхронізує зміни локальних даних з віддаленими оновленнями, які відбулися, коли клієнт був в автономному режимі, автоматично об'єднуючи будь-які конфлікти.

База даних Realtime надає гнучкий мову правил на основі виразів, званий Firebase Realtime Database Security Rules, для визначення того, як ваші дані повинні бути структуровані і коли дані можуть бути прочитані або записані. При інтеграції з Firebase Authentication розробники можуть визначати, хто має доступ до яких даними і як вони можуть отримати до них доступ.

База даних реального часу - це база даних NoSQL, тому вона відрізняється оптимізацією і функціональністю в порівнянні з реляційною базою даних. API бази даних реального часу призначений тільки для операцій, які можуть виконуватися швидко.

Це дозволяє вам створювати відмінні умови для роботи в реальному часі, які можуть обслуговувати мільйони користувачів без шкоди для швидкості реагування. У зв'язку з цим важливо подумати про те, як користувачі повинні отримувати доступ до ваших даних, а потім відповідним чином їх структурувати.

2.3.4 Вибір засобу візуалізації

Для візуалізації було вирішено використовувати SCSS,HTML. Відмінність SCSS від звичайного CSS більш розширеним функціоналом. Можливість створювати змінні, міксіни, наслідування та вкладеність.

Міксіни справді є ключовими для нашого розвитку. Вони дозволяють створювати багаторазові блоки. Вони дозволяють зробити кілька речей: розділити функцію на кілька, визначити, де житимуть блоки у межах ваших функцій, і змінити порядок, в якому вони житимуть.

Розділ 3

Програмна реалізація

3.1 Структура і функціональне призначення модулів системи

Структура модулів складається з звичайних React компонентів. Для прикладу розглянемо нижче наведений код на якому зображено синтаксис ES6, Функціональний компонент React PDF[20]. В даному прикладі показана не звичайна для JS практика - взаємодія з файловою системою. Також можна спостерігати створення нових екземплярів класу, що рідко використовуються при функціональному підході.

Даний модуль відповідає за виведення результуючого файлу.

```
You, seconds ago | 1 author (You)
1  import React from "react";
2  import { jsPDF } from "jspdf";
3  import html2canvas from "html2canvas";
4  import { Document } from "./document";
5
6  export const PDF = () => {
7    const printDocument = () => {
8      const input = document.getElementById("divToPrint");
9
10     html2canvas(input).then((canvas) => {
11       const imgData = canvas.toDataURL("image/png");
12
13       const pdf = new jsPDF();
14
15       pdf.addImage(imgData, "PNG", 0, 0);
16
17       const date = Date.now();
18
19       input.save(`CalculatorEstimate${date}.pdf`);
20     });
21   };
22
23   return (
24     <div className="m" style={{ height: "50px", overflow: "hidden" }}>
25       <div className="mb5" style={{ height: "50px" }}>
26         <button onClick={printDocument}>Print</button>
27       </div>
28       <Document />
29     </div>
30   );
31 };
32
```

```

<Route to="/application-calculator/steps/">
  <div className="applicationCalcStep">
    <div className="applicationCalcMain">
      <Switch>
        {state.categories.map(
          (category) =>
            category.id === state.categoryId &&
            category.availableSteps.map((route, index) => {
              const path = `/application-calculator/steps/step-${
                index + 1
              }`;
              return (
                <Route
                  path={path.toString()}
                  render={(props) => (
                    <StepView
                      id={index + 1}
                      steps={category.availableSteps}
                    />
                  )}
                />
              );
            )}
        )}
      <Route
        path="/application-calculator/design"
        component={StepDesignView}
      />
    </div>
  </div>
</Route>

```

Наступним кодом зображення логіка генерації роутів, які створюються динамічно в залежності від обраної категорії. Важливою частиною реалізації є сам модуль підрахунку вартості продукту. Як видно на рис.3.3.3 це модуль який варіативно від введених даних виводить результати, що вираховуються за формулами наведеними нижче.

3.2 Розробка програмних модулів

Важливим аспектом роботи усіх модулів є їхня динамічність. Оскільки розроблена система дуже гнучка може використовуватись в будь-яких сферах, в залежності ввід введених адміністратором даних.

Реалізацію динамічного створення сторінок та маршрутизації по цим сторінкам зображено в коді нижче.

```
src > JS ScrollToTop.js > ...
You, a month ago | 1 author (You)
1  import { useEffect } from 'react';
2  import { withRouter } from 'react-router-dom';
3
4  const ScrollToTop = ({ history }) => {
5    useEffect(() => {
6      const unlisten = history.listen(() => {
7        window.scrollTo(0, 0);
8      });
9      return () => {
10       unlisten();
11     };
12   }, [history]);
13
14   return null;
15 };
16
17 export default withRouter(ScrollToTop);
18
```

```

    component={StepGetEstimateView}
  />
  {!isLoading &&
state.categoryIsSelected === false &&
history.location.pathname !== "/application-calculator" ? (
  history.push(`/application-calculator`)
) : (
  <Route to="/application-calculator/steps/">
    <div className="applicationCalcStep">
      <div className="applicationCalcMain">
        <Switch>
          {state.categories.map(
            (category) =>
              category.id === state.categoryId &&
              category.availableSteps.map((route, index) => {
                const path = `/application-calculator/steps/step-${
                  index + 1
                }`;
                return (
                  <Route
                    path={path.toString()}
                    render={(props) => (...
                )}
              )}
            )}
        </Switch>
      </div>
    </div>
  </Route>
);
)}

```

Як видно на коді, що зображений нижче розробку велась на мові програмування Java Script, з використанням бібліотеки React, в даному додатку не вдалось обійтись без використання додаткових бібліотек таких як Redux/MobX.[10]

В нижче наведено код зображену роботу модуля що відповідає за роботу пагінації, та навігації по сторінках.

```
useEffect(() => {
  setFeatureTime(0);
  const step =
    state.selectedSteps.length &&
    state.selectedSteps.find((step) => step.id === id);
  step && state.selectedSteps.length
  ? step.features.map((feature) => {
    feature.isSelected
    ? setFeatureTime((prevState) => {
      return (
        parseFloat(prevState) +
        parseFloat(calculateTime(state, feature))
      );
    })
    : setFeatureTime((prevState) => parseFloat(prevState));
  })
  : setFeatureTime(0);
}, [id, state]);

return (
  <p style={{ cursor: 'pointer' }} className="asideBarDropDownPrice">
    {parseFloat(featureTime).toFixed(1).replace(/.0+$/, '')}
  </p>
);
```

3.3 Тестування інформаційної системи

Тестування застосунку проводилось шляхом аналізу її зручності та функціональності в процесі звірки вимог на відповідність, що були поставлені в мету розробки системи. Дослідження проходить шляхом моніторингу за системними

процесами в ситуаціях, що були штучно створені, та з обмеженим набором тестових випадків.

Для аналізу функціональності вибору платформи було розроблено тестовий випадок (тест-кейс), що перевіряє процес вибору платформи у застосунку.[11] При переході на сторінку вибору платформи реалізовано автоматичний вибір однієї з платформ для запобігання продовження виконання роботи застосунку без обраної платформи.(рис. 3.3.1).

Таблиця 3.1 – Тест-кейс АТ0001

Тест-кейс ID: АТ0001	Пріоритет: 1	Створено: 23.05.2021, В. Дасига
Назва: Перевірка вибору платформи		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> 1. Запустити додаток 2. Вибрати необхідну сферу застосування 3. Натиснути кнопку «SAVE AND CONTINUE» 4. Порівняти фактичний результат з очікуваним 	<p>Автоматично обирається одна з платформ</p> <p>Сторінка вибору платформи закрилась. Сторінку для вибору основних функціональних можливостей відкрито.</p>	
Результат виконання тест-кейсу: пройдено успішно		

The screenshot displays a user interface for selecting an app platform. At the top left, there is a 'Back' button. The main heading asks 'What type of app are you building?'. Below this, three options are presented: 'Android' with the Android logo, 'iOS' with the Apple logo, and 'Both' with both logos. The 'Both' option is highlighted with a grey background. A blue 'SAVE AND CONTINUE' button is positioned below the options. On the right side, a summary panel shows time estimates: 'SUMMARY TIME' at 29.8h, 'DEVELOPMENT TIME' at 19.9h, and 'NON-DEV TIME' at 9.9h. At the bottom, a 'Platform:' section shows 'Both' selected, and a 'Features:' section shows 'GPS Location' with a close button.

Category	Time
SUMMARY TIME	29.8h
DEVELOPMENT TIME	19.9h
NON-DEV TIME	9.9h

Рисунок 3.3.1 – Автоматичний вибір, якщо платформа не обрана

3.4 Інструкція користувача

Перш за все потрібно обрати сферу застосування яка найбільше підходить для майбутнього продукту (рисунок 3.4.1). Кожна зі сфер має певні унікальні функціональні можливості притаманні для неї.

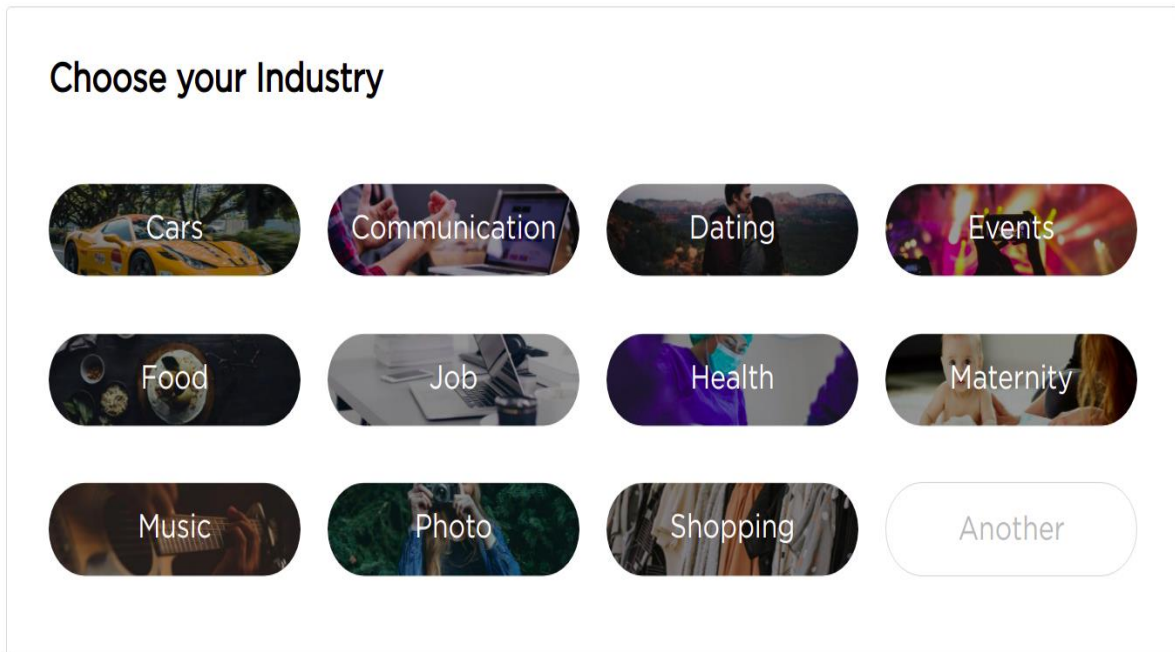


Рисунок 3.4.1 – Екран вибору сфери

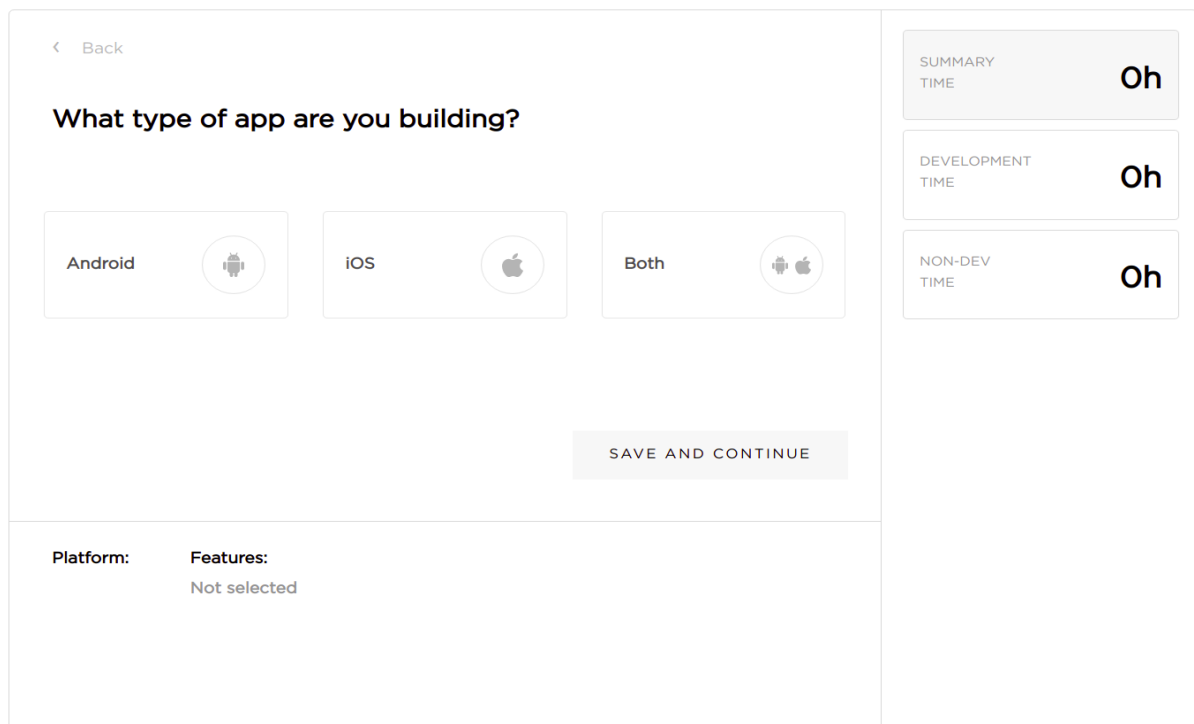


Рисунок 3.4.2 – Екран вибору платформи

Після вибраної індустрії з'явиться вікно де потрібно обрати платформу на якій буде зроблений програмний продукт. Це може бути Android, IOS, або обидві платформи.

The screenshot displays a configuration screen for a mobile application. At the top left, there is a back arrow and the text "Back". At the top right, it indicates "STEP 2 OFF 9". The main heading is "Will you send Notifications to your users?". Below this heading are two buttons: "Yes +43h" and "No". To the right of the main content area, there is a summary section with three items: "SUMMARY TIME 91.2h", "DEVELOPMENT TIME 57.2h", and "NON-DEV TIME 34h". At the bottom of the screen, there are two sections: "Platform:" with a button for "Android" and "Features:" with a button for "Registration & Authorization" that has an 'X' icon next to it, indicating it is selected.

Category	Value
SUMMARY TIME	91.2h
DEVELOPMENT TIME	57.2h
NON-DEV TIME	34h

Рисунок 3.4.3 – Екран вибору функціональної можливості

Отже, після цього з'являтиметься ряд питань чи потрібна певна можливість, чи ні, який одразу буде підраховувати робочі години у таблиці справа.

При виборі так на одне з питань потрібно чек-боксами проставляти нові функції або забрати вже ті які стоять за замовчуванням.

← Back STEP 1 OFF 9

Do you need Registration & Authorization?

Please chose the features below:

Enough for MVP	Additional Features
<input checked="" type="checkbox"/> Integration 1.5 hours	<input type="checkbox"/> via Phone 22 hours
<input checked="" type="checkbox"/> via Email 27 hours	<input type="checkbox"/> via Facebook 22.5 hours
<input checked="" type="checkbox"/> Add payment details 15 hours	<input type="checkbox"/> via Twitter 14 hours
<input checked="" type="checkbox"/> Forgot password 8 hours	<input type="checkbox"/> via Google+ 19 hours

SAVE AND CONTINUE

Platform: Android Features: X Registration & Authorization

SUMMARY TIME **91.2h**

DEVELOPMENT TIME **57.2h**

NON-DEV TIME **34h**

Рисунок 3.4.4 – Екран вибору додаткових складових функціональної можливості

При нажаті на особливості проекту(features, можеш фічі назвати) є можливість відразу перейти на додаткові функції та виправити щось. Це є дуже зручно, щоб не шукати та не повертатися назад, для більш зручної навігації по калькулятору.

BACK

Rough Estimate of My App

SUMMARY TIME 122.6h	OPTIMISTIC 110.3h
	PESSIMISTIC 134.9h

Estimated release date 11 May 2021. Leave request to start a project in 2 days.

DEVELOPMENT PRICE \$ 832	MINIMUM \$ 693
	MAXIMUM \$ 971

YOUR DEDICATED TEAM

- Android developer
- ✓ ios developer
- ✓ Web developer
- ✓ Project Manager
- ✓ QA Engineer
- ✓ UI/UX Designer

Leave your **email** to download your **Rough Estimate** with full Features list

Your Email

GET MY ESTIMATE

Рисунок 3.4.5 – Екран відображення результатів

В кінці так само обов'язковим рішенням є вибір дизайну, від нього також залежить кількість годин.

← Back

STEP 10 OFF 10

Do you need a design for your App?

None +0

Simple +12

Custom +21

GET ESTIMATE

SUMMARY TIME	122.6h
DEVELOPMENT TIME	83.2h
NON-DEV TIME	39.4h

Platform: ios

Features: X Registration & Authorization

Рисунок 3.4.6 – Екран вибору дизайну

Сторінка яку бачить користувач після пройденого етапу вибору функцій свого майбутнього застосунку. На цій сторінці можливо одразу побачити коротку інформацію про майбутній проект скільки само він буде коштувати та скільки часу займе час його розробки.

Також нажавши Get My Estimate є можливість скачати повний файл.

Registration & Authorization	
Integration	5.0h
via Email	38.0h
via Twitter	19.0h
via Google+	26.0h
Payment gateway	
PayPal	58.1h
Non development time	
Project setup	10.9h
Architecture	8.5h
Database	7.9h
Swagger	6.9h
Network	2.6h
Meetings	5.6h
Continuous integration	5.2h
Continuous delivery	2.6h
Release per sprint	11.7h
Release per integration	2.6h
Backend:	\$580
Frontend:	\$50
Design:	\$90
Mobile:	\$741
Management:	\$162
Total:	\$1623USD

Рисунок 3.4.7 – Файл відображення результатів

Оразу після завантажування файлу є можливість побачити розрахунок проекту обширно, включаючи розпис усього вибраного функціоналу.

3.5 Вимоги до розгортання інформаційної системи

Для розгортання готової операційної системи бажано використовувати комп'ютер, що має наступні технічні характеристики.

1. Процесор: з тактовою частотою 2 ГГц (допускаються потужніші), з підтримкою технологій NX, PAE і SSE2;
2. Оперативна пам'ять: 4 ГБ (для 32-розрядної версії операційної системи) або 6 ГБ (для 64-розрядної версії операційної системи);

3. Вільний простір на диску: 1 МБ (для 32-розрядної версії) або 1 МБ (для 64-розрядної версії);
4. Графічна плата: графічний пристрій Microsoft DirectX 9 із драйвером WDDM;
5. Необхідний доступ до інтернету необхідний так як застосунок буде розортатись в веб просторі (може стягуватися плата за послуги провайдера).

Також потрібно впевнитися в забезпеченні необхідних розширень та програм, таких як:

Рекомендовані браузері:

- веб-браузер Microsoft Internet Explorer версія 6.x і вище;
- веб-браузер Google Chrome версія 3.0.194 і вище;
- веб-браузер Mozilla Firefox версія 3.1 і вище;
- веб-браузер Safari версія 3.1 і вище.

Вимоги до сервера:

- Node JS, для розгортання самого сервера;
- Встановленні всі залежності згідно до package.json;
- Ngrok для запуску додатку в мережу (в комерційних цілях, доцільніше використовувати хостинги);
- Sass компілятор, для компіляції SCSS в готовий CSS код.

Перелік посилань

1. Застосування інформаційних технологій. URL:
<https://www.bsmu.edu.ua/blog/2525-zastosuvannya-info-tehnologii-ta-ih-rol/>
2. База даних Firebase в реальному часі. URL:
<https://firebase.google.com/docs/database>
3. .NETFramework. URL: https://uk.wikipedia.org/wiki/.NET_Framework
4. PHP. URL: <https://uk.wikipedia.org/wiki/PHP>
5. HTML — мова розмітки гіпертексту. URL: <http://www.znannya.org/?view=html>
6. React. URL: <https://uk.wikipedia.org/wiki/React>
7. React. URL: <https://www.wikizero.com/uk/React>
8. Мова JavaScript та її можливості. URL:
<https://www.sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-jeie-mozlivosti>
9. UI-фреймворки та бібліотеки JavaScript. URL: <https://codeguida.com/post/2322>
10. Документація SASS URL: <https://sass-scss.ru/documentation/>
11. What Is Windows Communication Foundation URL: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>
12. React Java Script бібліотека для створення інтерфейсів URL:
<https://ru.react.js.org/>
13. Що таке SCSS URL:
<https://habr.com/ru/post/140612/#:~:text=SCSS%20%E2%80%94%20%D0%B4%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%22%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0%20SASS,%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D1%82%D1%81%D1%8F%20%D0%B2%20%D0%BE%D0%B1%D1%8B%D1%87%D0%BD%D1%8B%D0%B9%20CSS%20%D0%BA%D0%BE%D0%B4.>
14. React Java Script бібліотека для створення інтерфейсів URL:
<https://ru.react.js.org/>

15. Початок роботи з React URL:

https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started

16. NOSQL переваги та недоліки нереляційних баз даних URL:

<https://www.quality-assurance-group.com/nosql-perevagy-ta-nedoliky-nerelyatsijnyh-baz-danyh/>

17. Різниця між базою даних NoSQL і SQL URL: <https://uk.natapa.org/difference-between-nosql-and-sql-database-872>

18. Оптимізація продуктивності URL: <https://uk.reactjs.org/docs/optimizing-performance.html>

19. Використовуємо хук ефекту URL: <https://uk.reactjs.org/docs/hooks-effect.html>

20. Portable Document Format URL:

https://ru.wikipedia.org/wiki/Portable_Document_Format

Висновки

В результаті виконання кваліфікаційної роботи бакалавра на тему веб застосунок для підрахунку собівартості товарів та послуг було отримано готовий програмний продукт, який задовільняє всі сучасні вимоги в програмуванні. Всі бізнес-процеси виконують своє пряме призначення, та забезпечують надійну роботу даної системи.

Сам програмний продукт повністю виконує свої поставленні задачі. В процесі реалізації було обрано сферу надання послуг в області інформаційних технологій, з метою висвітлення етапів побудови програмних продуктів, етапів проектування та процесу ціноутворення з можливістю кореляції даного продукту на етапі його замовлення.

Мета розробки комплексної роботи бакалавра, є веб застосунок для підрахунку собівартості товарів та послуг, який повинний забезпечити інформативність майбутніх клієнтів на етапі вирішення, який програмний продукт вони хочуть отримати.

Розробка буде виконувалась за допомогою бібліотеки React, з використанням не реляційної БД Firebase Realtime Database.

Інформаційні технології - це інструмент, що відповідає інформаційну діяльність та її роботу, пов'язану з підготовкою та прийняттям рішень.

Отже, звідси можна зробити висновок, що використання інформаційних технологій з метою покращення інформативного забезпечення клієнтів, покращить досвід співпраці з компанією, та принесе відкритість та розуміння формування цін даною компанією.

ДОДАТКИ

Додаток А

**Програмні коди основних модулів веб застосунку для підрахунку
собівартості продуктів**

Лістинг applicationCalcDesignView

```

@import "../settings/mixins";

.applicationCalcStepDesign {
display: flex;TTimes New Roman
justify-content: space-between;
height: 740px;
}
.stepDesignContainer {
height: 522px;
width: 100%;
}
.stepHeader {
margin-top: 20px;
margin-left: 40px;
h2 {
font-family: "GothamPro", GothamPro, sans-serif;

margin: 0;
}
}
.stepDesignContentDiv {
@include containerCreator;
opacity: 0.7;

.designPlusTime {
color: $textAlternativeColor1;
}
}
.stepDesignContentDivSelected {
@include containerCreator;
@include contAnimation;
}
.stepDesignContentDiv:hover {
@include containerCreatorHover;
}
.stepDesignSelector {
height: 43%;
margin-top: 80px;
display: flex;
justify-content: space-evenly;
}
.stepDesignContentNone,
.stepDesignContentSimple,
.stepDesignContentCustom {
@include roundContainerForIcon;
}
.stepDesignContentNone {
@include icon;
background-image: url("../images/icon/design/none.svg");
}

```

```

}
.stepDesignContentSimple {
@include icon;
background-image: url("../images/icon/design/simple.svg");
}
.stepDesignContentCustom {
background-image: url("../images/icon/design/simple.svg"),
url("../images/icon/design/custom.svg"),
url("../images/icon/design/none.svg");
background-repeat: no-repeat;
background-position: right 30% bottom 40%, left 50% top 25%,
left 30% bottom 40%;
background-size: 20px;
}

.stepDesignTimeIcon {
display: flex;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

margin-right: 10px;
}
}

```

ЛІСТИНГ applicationCalcMain

```

@import "../applicationCalcOne";
@import "../applicationCalcStepTwo";
@import "../asideBar";
@import "../bottomBar";
@import "../stepBack";
@import "../stepCounter";
@import "../saveAndContinue";
@import "../settings/mixins";
@import "../elements/selectorCheckBox";
@import "../applicationCalcDesignView";
@import "../estimateView";
@import "../getEstimateView";
@import "../responsiveStyles";
.overlay {
display: none;
transition: 2s;
}
.applicationCalc {
border: 1px solid #d8d8d8;
border-radius: 5px;
max-width: 1120px;
min-width: 320px;
height: 100%;
overflow: hidden;
margin: auto 0;
}
.stepHeaderDiv {
height: 80px;
display: flex;
flex-direction: row;
justify-content: space-between;

```

```

}
.applicationCalcMain {
width: 820px;
}
.applicationCalcStep {
display: flex;
justify-content: flex-start;
}
.featureAddButton {
width: 230px;
height: 110px;
border: 1px solid rgba(148, 148, 148, 0.5);
border-radius: 4px;
display: flex;
justify-content: space-evenly;
margin-right: 20px;
transition: 0.3s ease-in-out;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

opacity: 0.5;

text-align: center;
margin: auto;
margin-left: 20px;
}
h2 {
font-family: "GothamPro", GothamPro, sans-serif;

opacity: 0.5;

text-align: center;
margin: auto;
margin-right: 20px;
margin-right: 32px;
font-weight: 100;
}
}
.featureAddButton:hover {
transition: 0.3s ease-in-out;
background-color: $disabledBackground;
}
.featureAddContainer {
margin-bottom: 40px;
height: max-content;
margin-left: 40px;
margin-top: 30px;
display: flex;
justify-content: flex-start;
box-shadow: inset 0px 10px 200px -20px rgb(255, 255, 255);
.featureContentSubmitContainer {
margin-top: 10px;
margin-right: 210px;
float: right;
}
}
}

```

ЛІСТИНГ applicationCalcOne

```
.applicationCalcStepOneHeader {  
margin-left: 40px;  
margin-top: 40px;  
h3 {  
font-family: $font-default;  
font-size: $h1-font-size;  
  
margin: 0;  
}  
}  
.applicationCalcStepOneBody {  
margin: 40px;  
font-family: $font-default;  
font-size: $h3-font-size;  
display: flex;  
flex-wrap: wrap;  
justify-content: space-between;  
}  
.applicationCalcStepOneImg {  
position: absolute;  
margin: 0;  
width: 100%;  
height: 100%;  
}  
  
.applicationCalcStepOneImgText {  
width: 100%;  
z-index: 100;  
position: absolute;  
color: $textButtonsWithBackground;  
margin: 0;  
line-height: 70px;  
text-align: center;  
font-size: 25px;  
}  
.applicationCalcStepOneImgContainer {  
cursor: pointer;  
position: relative;  
overflow: hidden;  
margin: 21px 0;  
width: 240px;  
height: 70px;  
border-radius: 35px;  
}  
.applicationCalcStepOneAnotherText {  
width: 100%;  
z-index: 100;  
position: absolute;  
color: $textButtonsWithBackground;  
margin: 0;  
line-height: 70px;  
text-align: center;  
font-size: 25px;  
font-weight: 500;  
color: #b9b9b9;
```

```

}
.applicationCalcStepOneAnotherContainer {
  cursor: pointer;
  position: relative;
  overflow: hidden;
  margin: 21px 0;
  width: 240px;
  height: 70px;
  border-radius: 35px;
  border: solid 1.5px #d8d8d8;
  transition: 0.5s;
}
.applicationCalcStepOneAnotherContainer:hover {
  background-color: $secondaryBackgroundColor;
  transition: 0.5s;
}
.applicationCalcStepOneImgContainer .applicationCalcStepOneImg {
  transition: transform 0.5s;
  transform: scale(1);
}

.applicationCalcStepOneImgContainer:hover .applicationCalcStepOneImg {
  transition: transform 0.5s;
  transform: scale(1.1);
}

.applicationCalcStepOneSubmit {
  text-indent: 0;
  transition: text-indent 2s linear;
  font-size: 21px;
  cursor: pointer;
  margin: 21px 0;
  text-align: center;
  line-height: 70px;
  width: 240px;
  height: 70px;
  border-radius: 35px;
  background-color: $textButtonsWithBackground;
  color: $textSecondaryColor;
  border: 1.5px solid $borderPrimaryColor;
}

.applicationCalcStepOneSubmit:hover {
  background-color: $disabledBackground;
  transition: 0.5s;
}

```

ЛІСТИНГ applicationCalcStepTwo

```
@import "../settings/mixins";
```

```

.applicationCalcStepTwo {
  display: flex;
  justify-content: space-between;
  height: 740px;
}
.stepTwoContainer {

```

```

height: 522px;
width: 100%;
}
.stepHeader {
margin-top: 20px;
margin-left: 40px;
h2 {
margin: 0;
}
}
.stepTwoContentDiv {
@include containerCreator;
opacity: 0.7;
}
.stepTwoContentDivSelected {
@include containerCreator;
@include contAnimation;
}
.stepTwoContentDiv: hover {
@include containerCreatorHover;
}
.stepTwoPlatformsSelector {
height: 43%;
margin-top: 80px;
display: flex;
justify-content: space-evenly;
}
.stepTwoContentAndroid,
.stepTwoContentIOS,
.stepTwoContentBoth {
@include roundContainerForIcon;
}
.stepTwoContentAndroid {
@include icon;
background-image: url("../images/icon/android/grey.svg");
}
.stepTwoContentIOS {
@include icon;
background-image: url("../images/icon/ios.svg");
}
.stepTwoContentBoth {
background-image: url("../images/icon/android/grey.svg"),
url("../images/icon/ios.svg");
background-repeat: no-repeat;
background-position: left 25% bottom 50%, right 25% bottom 50%;
background-size: 15px;
}
.stepTwoContentDivSelected .stepTwoContentAndroid {
background-image: url("../images/icon/android/green.svg");
}
.stepTwoContentDivSelected .stepTwoContentIOS {
background-image: url("../images/icon/iosBlue.svg");
}
.stepTwoContentDivSelected .stepTwoContentBoth {
background-image: url("../images/icon/android/green.svg"),
url("../images/icon/iosBlue.svg");
}

```

ЛІСТИНГ `asideBar`

```

.asideBarContainer {
padding-top: 10px;
box-sizing: border-box;
width: 300px;
border-left: 1px solid $borderPrimaryColor;
position: relative;
}
.asideBarContentMain {
border-radius: 4px;
margin: 10px 20px;
border: 1px solid $borderPrimaryColor;
width: 260px;
min-height: 90px;
max-height: 340px;
overflow: hidden;
.asideBarContent {
display: flex;
justify-content: space-evenly;
height: 90px;
transition: 0.3s ease-in-out;
h5 {
font-family: 'GothamPro', GothamPro, sans-serif;
cursor: pointer;
}
h2 {
font-size: $h1-font-size;
font-family: 'GothamPro', GothamPro, sans-serif;
cursor: pointer;
width: 55%;
text-align: right;
margin-right: 15px;
margin-bottom: 25px;
margin-top: auto;
}
.asideBarContentHeader {
text-transform: uppercase;
color: $secondaryBackgroundColor;
line-height: 22px;
font-weight: 500;
width: 45%;
margin-left: 15px;
margin-bottom: 25px;
margin-top: auto;
}
}
}

.asideBarContentMain: hover {
transition: 0.3s ease-in-out;
background-color: $disabledBackground;
}

.asideBarDropDownContent {
padding-bottom: 10px;

```

```

overflow-y: scroll;
max-height: 240px;

.asideBarDropDownContainer {
display: flex;
justify-content: space-between;
padding-left: 20px;
padding-right: 20px;
color: $textPlatformsAndFeaturesColor;
font-size: 12px;
p {
font-family: 'GothamPro', GothamPro, sans-serif;
cursor: pointer;
margin-top: 10px;
margin-bottom: 10px;
}
.asideBarDropDownPrice {
margin-top: 10px;
margin-bottom: 10px;
font-weight: 700;
}
}
.asideBarContentDescription {
position: absolute;
border: 1px solid $borderPrimaryColor;
border-radius: 4px;
width: 259px;
min-height: 94px;
font-size: 12px;
color: $textPlatformsAndFeaturesColor;
margin: 0px 20px;
display: flex;
justify-content: center;
p {
font-family: 'GothamPro', GothamPro, sans-serif;
line-height: 18px;
margin: 0;
padding: 20px;
}
}

```

Лістинг bottomBar

```

.bottomBarContainer {
height: 215px;
width: 820px;
border-top: 1px solid $borderPrimaryColor;
display: flex;
justify-content: flex-start;
flex-direction: row;
}
.platform {
width: 130px;
height: 100%;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

margin-left: 40px;
margin-top: 30px;
}
}
.platformContent {
margin: 10px 24px;
width: 80%;
height: 32px;
opacity: 0.5;
border-radius: 16px;
border: solid 1px $textSecondaryColor;
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

margin-top: 9px;
text-align: center;
color: $textPlatformsAndFeaturesColor;
font-weight: normal;
}
}
.features {
overflow-y: hidden;
margin-left: 40px;
width: fit-content;
height: 100%;
justify-content: flex-start;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

margin-top: 30px;
}
}
.featuresContent {
margin: auto;
display: flex;
justify-content: space-between;
height: 32px;
width: max-content;
opacity: 0.5;
border-radius: 0px 16px 16px 0px;
border: solid 1px $textSecondaryColor;
border-left: 0;
padding-left: 8px;
padding-right: 12px;
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

width: fit-content;
text-align: center;
margin-top: 9px;
color: $textPlatformsAndFeaturesColor;
font-weight: normal;
}
}
.featuresRemove {
border-radius: 16px 0px 0px 16px;

```

```

background-image: url("../images/icon/close/grey.svg");
background-position: center;
background-repeat: no-repeat;
background-size: 16px;
background-color: $disabledBackground;
width: 40px;
height: 100%;
}
.featuresContainer {
display: flex;
justify-content: flex-start;
flex-wrap: wrap;
overflow: scroll;
height: 100%;
padding-bottom: 20%;
}
.featuresRemove:hover {
background-color: $secondaryBackgroundColor;
}
.featuresRemove:active {
background-color: $primaryBackground;
}
.featuresBody {
display: flex;
height: 32px;
margin-right: 20px;
margin-top: 10px;
}
.progress {
height: 3px;
background-color: black;
margin-top: -1.5px;
border-radius: 0% 3px 3px 0%;
}
.progress-div {
position: absolute;
width: 820px;
height: 1px;
}
.notSelectedFeatures {
font-size: 16px;
color: #949494;
margin-top: 15px;
}

```

ЛІСТИНГ estimateView

```

.releaseDate {
margin-top: 30px;
background-color: #f7f7f7;
border-radius: 5px;
width: 450px;
height: 39px;
font-size: 11px;
line-height: 18px;
text-align: left;
text-align: center;
}

```

```

vertical-align: bottom;
padding-top: 10px;
}
.summaryTimeEstimateContainer {
background-color: #ededed;
border-radius: 4px;
width: 450px;
height: 130px;
display: flex;
justify-content: space-between;
.summaryTimeEstimateMain {
width: 246px;
border-right: 1px solid white;
height: 100%;
h4 {
font-family: 'GothamPro', GothamPro, sans-serif;

```

```

margin-left: 20px;
margin-top: 29px;
}
h3 {
font-family: 'GothamPro', GothamPro, sans-serif;

```

```

margin-left: 20px;
margin-top: 25px;
}
}
.summaryTimeEstimateAsideContainer {
.top {
border-bottom: 1px solid white;
}
.summaryTimeEstimateAsideItem {
height: 65px;
width: 203px;
}
}
}

```

```

.developmentPriceEstimateContainer {
margin-top: 30px;
border: 1px solid #ededed;
border-radius: 4px;
width: 450px;
height: 130px;
display: flex;
justify-content: space-between;
.developmentPriceEstimateMain {
width: 246px;
border-right: 1px solid #ededed;
height: 100%;
h4 {
font-family: 'GothamPro', GothamPro, sans-serif;

```

```

margin-left: 20px;
margin-top: 29px;
}
h3 {
font-family: 'GothamPro', GothamPro, sans-serif;

```

```

margin-left: 20px;
margin-top: 25px;
}
}
.developmentPriceEstimateAsideContainer {
.top {
border-bottom: 1px solid rgb(237, 237, 237);
}
.developmentPriceEstimateAsideItem {
height: 65px;
width: 203px;
}
}
}
.EstimateAsideItemHeader {
padding-top: 15px;
padding-left: 15px;
}
.EstimateAsideItemValue {
padding-top: 7px;
padding-left: 15px;
}
.estimateHeader {
margin-top: 29px;
margin-left: 40px;
}
.stepEstimateContainer {
height: 594px;
}
.summaryTimeAndDevelopmentPrice {
width: 450px;
margin-left: 40px;
}
.dedicatedTeam {
border: 1px solid rgb(216, 216, 216, 0.5);
border-radius: 4px;
width: 265px;
height: 359px;
color: #000000;
h4 {
font-family: 'GothamPro', GothamPro, sans-serif;

margin-top: 20px;
margin-left: 20px;
}
}
.getEstimateContainer {
border: 1px solid rgb(216, 216, 216, 0.5);
border-radius: 4px;
width: 265px;
height: 359px;
display: flex;
justify-content: space-evenly;
flex-direction: column;
}
.estimateMainContainer {

```

```

margin-top: 50px;
display: flex;
justify-content: space-between;
flex-wrap: wrap;
margin-right: 40px;
}
.dedicatedTeamList {
ul {
padding-left: 0px;
margin-left: 20px;
list-style: none;
li {
margin-top: 10px;
input[type='checkbox'] {
display: none;
}
input[type='checkbox']:checked + div {
width: 15px;
height: 15px;
background-image: url('../images/checkmark.svg');
}
input[type='checkbox'] + div {
width: 15px;
height: 15px;
background-color: white;
}
label {
display: flex;
flex-wrap: wrap;
justify-content: flex-start;
}
div {
margin-right: 10px;
}
}
}
}
.getEstimateButton {
background-color: #ededed;
width: 225px;
height: 50px;
font-size: 14px;
letter-spacing: 1.5px;
border: 0;
color: #000000;
a {
color: black !important;
}
}
.getEstimateButtonHolder {
display: flex;
justify-content: center;
}
.getEstimateEmailText {
color: #000000;
font-family: GothamPro;
font-size: 12px;

```

```

line-height: 18px;
width: 183px;
text-align: center;
}
.getEstimateEmailTextContainer {
display: flex;
justify-content: center;
}
.getEstimateEmail {
border: 0;
border-bottom: 1.5px solid #d8d8d8;
width: 225px;
color: #000000;
font-size: 12px;
line-height: 38px;
}

```

Лістинг getEstimateView

```

.stepGetEstimateContainer {
border: 1px solid #d8d8d8;
border-radius: 5px;
width: 1120px;
height: 593px;
display: flex;
justify-content: flex-start;
flex-wrap: wrap;
flex-direction: column;
h2 {
font-family: 'GothamPro', GothamPro, sans-serif;

margin-top: 110px;
text-align: center;
}
h5 {
font-family: 'GothamPro', GothamPro, sans-serif;

text-align: center;
}
img {
margin-top: 60px;
}
div {
margin: 0 auto;
width: 225px;
}
}

```

Лістинг responsiveStyles

```

@media screen and (min-width: 576px) {
.featureContent {
flex-direction: row !important;
}
}

@media screen and (max-width: 1118px) {
::-webkit-scrollbar {

```

```

width: 0px;
}

body {
margin: 0px;
}
.root {
min-width: 320px;
width: min-content;
}
.applicationCalcStepOneImgContainer .applicationCalcStepOneImg,
.applicationCalcStepOneImgContainer:hover .applicationCalcStepOneImg,
.applicationCalcStepOneAnotherContainer,
.applicationCalcStepOneAnotherContainer:hover {
transform: none;
transition: none;
background-color: inherit;
}

.asideBarContainer {
padding-top: 0px;
transition: 1s;
background-color: white;
position: fixed;
top: 0;
bottom: 0;
z-index: 3;
width: 100%;
transform: translateX(100%);
.quickViewButton {
width: 32px;
height: 100px;
position: absolute;
top: 50%;
transform: translateY(-50%) translateX(-32px);
background-color: $yellowPrimary;
// border-radius: 5px 5px 5px 5px;
font-size: 14px;
flex-wrap: wrap;
box-shadow: 0px 0px 15px 1px #33cfff80;
// border: 1px solid #d8d8d8;
width: 32px;
transition: transform 1s;
padding-left: 2px;
padding-top: 10px;
.quickViewSideBarArrow {
margin: 0 auto;
width: 14px;
height: 14px;
background-image: url("../images/icon/arrow/black.svg");
background-position: center;
background-size: 14px 14px;
background-repeat: no-repeat;
transform: rotate(180deg);
transition: transform 1s;
}
p {

```

```
font-family: "GothamPro", GothamPro, sans-serif;
```

```
cursor: pointer;
margin: 7px !important;
transform: rotate(180deg);
margin: 0px;
writing-mode: vertical-lr;
}
}
}
```

```
.asideBarMainContainer {
margin-top: 74px;
}
```

```
.overlay {
display: none;
transition: 2s;
}
```

```
.quickView .overlay {
display: block;
width: 100%;
height: 100%;
position: fixed;
background-color: rgba(0, 0, 0, 0.3);
transition: 2s ease-in-out;
z-index: 2;
}
```

```
.quickView .quickViewButton {
position: fixed;
transition: 1s;
```

```
.quickViewSideBarArrow {
transform: rotate(360deg);
}
}
```

```
.quickView .quickViewButton {
transform: translateX(0px) translateY(-50%);
}
```

```
.quickView .asideBarContainer {
padding-top: 0px;
transition: 1s;
overflow-y: scroll;
transform: translateX(0%);
}
```

```
.quickView .asideBarContent {
width: 100% !important;
}
```

```
.asideBarContent {
transition: 1s;
}
```

```
.applicationCalc {
height: max-content;
min-width: 100%;
overflow: visible;
}
```

```
.applicationCalcMain {
width: 100%;
```

```

overflow: visible;
}
.applicationCalcStep {
width: 100%;
overflow: visible;
}
.featureContentSubmitContainer {
margin: 30px auto !important;
width: fit-content;
float: unset !important;
}
.featureContent {
flex-direction: column;
height: max-content;
}
.featuresSelectArea {
width: 80%;
height: max-content;
margin-right: 0px;
margin: 10px auto;
}
.featureContentMVP {
width: 100% !important;
}
.featureContentAdditionalFeatures {
width: 100% !important;
}
.stepTwoContainer {
height: max-content;
}

.stepDesignContainer {
height: max-content;
}
.saveAndContinueSubmit {
position: relative;
}
.platform {
margin-left: 20px;
height: min-content;
width: 100%;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

margin-left: 0px !important;
position: relative;
}
div {
margin-left: 0px !important;
}
}
.platformContent {
width: 110px;
margin: 20px 30px;
border: solid 1px rgba(216, 216, 216, 0.5);
opacity: 1;
}

```

```

.features {
margin-left: 20px;
overflow: visible;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

z-index: -1;
margin-top: 0px;
position: relative;
}
}

.bottomBarContainer {
flex-direction: column;
overflow-y: visible;
width: 100%;
height: 100%;
}
.applicationCalcStepOneImgContainer,
.applicationCalcStepOneAnotherContainer {
margin: 10px auto;
}
.stepTwoPlatformsSelector {
flex-wrap: wrap;
margin-top: 14px;
padding: 30px;
}
.stepHeader {
margin: 0 auto;
text-align: center;
h2 {
font-family: "GothamPro", GothamPro, sans-serif;

margin: 20px;
font-size: 22px;
}
}
.submitHolderContent {
animation: none;
}
.submitHolderElement {
display: none;
}
.saveAndContinueContainer {
float: none;
margin: 20px auto;
}
.progress-div {
width: 100%;
position: inherit;
}
.stepDesignSelector {
flex-wrap: wrap;
margin-top: 14px;
height: fit-content;
justify-content: center;
width: 100%;
}

```

```

padding: 30px;
}
.stepDesignContentDivSelected,
.stepDesignContentDiv {
margin: 10px 0px;
width: 100%;
}
.featureAddContainer {
flex-direction: column;
margin-left: 0 !important;
width: 100%;
padding: 30px;
div {
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

text-align: center;
}
}
}
.featureAddButton {
width: 100% !important;
margin-right: 0px !important;
margin: 10px auto !important;
border: 1px solid rgba(148, 148, 148, 0.5) !important;
}
.stepTwoContentDiv,
.stepTwoContentDivSelected {
width: 100%;
margin-bottom: 10px;
}

.stepHeaderDiv {
height: 50px !important;
}
.stepBackButton {
margin-top: 0px;
margin-left: 10px;
}
.stepCounter {
margin: auto 20px;
}
.featuresBody {
max-width: fit-content;
}
.featuresContainer {
margin-top: 20px;
}
.stepQuickView {
display: none;
}
.stepBackButton {
margin-left: 20px;
h2 {
font-family: "GothamPro", GothamPro, sans-serif;

margin-right: 10px;

```

```

}
}
.applicationCalcSelector {
h6 {
font-family: "GothamPro", GothamPro, sans-serif;

color: rgba(17, 17, 17, 0.5);
opacity: 1;
}
}
.asideBarContentMain,
.asideBarContentDescription {
width: 80%;
margin: 20px auto;
}
.asideBarContentDescription {
min-height: fit-content;
position: relative;
margin: 20px auto;
}
.estimateHeader {
margin: 20px auto;
text-align: center;
h2 {
font-family: "GothamPro", GothamPro, sans-serif;
font-size: 18px;
h2 {
text-align: center;
}
}
}
.stepEstimateContainer {
height: 100%;
}
.estimateMainContainer {
margin: 0px;
padding-left: 20px;
padding-right: 20px;
}
.summaryTimeAndDevelopmentPrice {
margin: 20px auto;
width: 100%;
display: flex;
justify-content: space-between;
flex-wrap: wrap;
}
.summaryTimeEstimateContainer,
.developmentPriceEstimateContainer {
width: 48%;
margin: 10px 0px;
}
.releaseDate {
width: 100%;
order: 3;
}

.dedicatedTeam,

```

```

.getEstimateContainer {
margin: 20px 0px;
width: 48%;
height: fit-content;
}
.dedicatedTeam {
padding-top: 40px;
padding-bottom: 40px;
}
.dedicatedTeamHeader {
text-align: center;
}
.dedicatedTeamList {
ul {
width: 200px;
display: flex;
justify-content: center;
flex-wrap: wrap;
margin: 20px auto;
li {
width: 200px;
}
}
}
.releaseDate {
height: fit-content;
padding: 10px;
}
.summaryTimeEstimateMain,
.summaryTimeEstimateAsideContainer,
.developmentPriceEstimateMain,
.developmentPriceEstimateAsideContainer {
width: 50% !important;
}
.developmentPriceEstimateAsideItem {
width: 100% !important;
}
.stepGetEstimateContainer {
width: 100%;
height: 100vh;
h2 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: 18px;
}
div {
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: 14px;
}
}
}
.applicationCalcStepOneHeader {
margin: 20px auto;
h3 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

width: 100%;
text-align: center;
}
}
}

@media screen and (min-width: 1118px) {
::-webkit-scrollbar {
width: 2px;
}

::-webkit-scrollbar-track {
}

::-webkit-scrollbar-thumb {
width: 2px;
background-color: #d8d8d8;
}
}

@media screen and (max-width: 375px) {
.summaryTimeEstimateContainer,
.developmentPriceEstimateContainer {
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: 13px;
}
}
.summaryTimeEstimateMain,
.developmentPriceEstimateMain {
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: 12px;
}
}
.summaryTimeEstimateAsideItem,
.developmentPriceEstimateAsideItem {
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: 12px;
}
}
}

@media screen and (max-width: 572px) {
.dedicatedTeam {
height: fit-content;
}
.estimateMainContainer {
display: flex;
justify-content: center;
}
}

@media screen and (min-width: 1118px) {

```

```

.quickViewButton {
display: none;
}
}
@media screen and (max-width: 681px) {
.summaryTimeEstimateContainer,
.developmentPriceEstimateContainer {
width: 100%;
}
.dedicatedTeam,
.getEstimateContainer {
min-width: 265px;
}
}
@media screen and (max-width: 571px) {
.dedicatedTeam,
.getEstimateContainer {
width: 100%;
}
.getEstimateContainer {
padding-bottom: 40px;
}
}
@media screen and (min-width: 571px) {
.dedicatedTeam,
.getEstimateContainer {
padding-bottom: 40px;
height: 395px;
}
}

```

ЛІСТИНГ saveAndContinue

```

.saveAndContinueContainer {
float: right;
margin: auto;
margin-right: 30px;
width: 260px;
height: 60px;
display: flex;
flex-direction: row;
overflow-x: hidden;
}

.saveAndContinueSubmit:hover {
transition: transform 0.2s;
transform: translateY(5px);
}

.saveAndContinueSubmit {
position: absolute;
transition: transform 0.2s;
transform: translateY(0px);
box-sizing: border-box;
border: 0;
width: 260px;
height: 50px;
}

```

```

background-color: $yellowPrimary;
font-size: $h5-font-size;
font-family: 'GothamPro', GothamPro, sans-serif;
letter-spacing: 2px;
}
.submitHolderElement {
@include buttonElementAlign;
width: 260px;
height: 50px;
div {
width: 260px;
height: 50px;
background-color: black;
}
}
.submitHolderContent {
@include buttonContentAlign;
width: 260px;
height: 60px;
}

```

ЛІСТИНГ selectorCheckBox

```

.featuresSelectArea {
height: 250px;
width: 700px;
overflow-y: scroll;
}
.applicationCalcSelector {
display: flex;
justify-content: flex-start;
input[type="checkbox"]:checked + .applicationCalcSelectorPseudo {
width: 18px;
height: 18px;
border-radius: 2px;
box-shadow: 0 10px 35px -15px #33cfff80;
border: 0;
background-color: #33cfff;
background-image: url("../images/checkmark.svg");
background-position: center;
background-size: 11px;
background-repeat: no-repeat;
}
h5 {
font-family: "GothamPro", GothamPro, sans-serif;

margin-top: 8px;
}
h6 {
font-family: "GothamPro", GothamPro, sans-serif;

margin: auto 15px;
color: #111111;
height: 12px;
opacity: 0.5;
font-size: 14px;
}

```

```

h6:after {
content: " hours";
}
}
.applicationCalcSelectorCheckbox {
display: none;
}
.applicationCalcSelectorPseudo {
margin: 5px;
width: 18px;
height: 18px;
border-radius: 2px;
box-shadow: 0 10px 35px -15px #33cfff80;
border: solid 1px #949494;
}
.featureContent {
display: flex;
justify-content: flex-start;
.featureContentMVP,
.featureContentAdditionalFeatures {
width: 50%;
}
}
}
.applicationCalcSelectorCheckboxNotEditable {
display: none;
}
}
.applicationCalcSelectorPseudoNotEditable {
margin: 5px;
width: 18px;
height: 18px;
border-radius: 2px;
box-shadow: 0 10px 35px -15px #33cfff80;
background-color: rgba(216, 216, 216, 0.5);
background-image: url("../images/checkmark.svg");
background-position: center;
background-size: 11px;
background-repeat: no-repeat;
}
}

```

Лістинг stepBack

```

.stepBackButton {
margin-left: 40px;
margin-top: 15px;
cursor: pointer;
width: 130px;
height: 50px;
display: flex;
flex-direction: row;
color: $textAlternativeColor;
h2 {
font-family: 'GothamPro', GothamPro, sans-serif;
margin: 0;
margin-right: 20px;
margin-top: 10px;
font-weight: 100;
}
}

```

```
p {
font-family: 'GothamPro', GothamPro, sans-serif;
margin: auto 0;
}
}
```

Лістинг stepCounter

```
.stepCounter {
margin: auto 40px;
width: 120px;
height: 15px;
display: flex;
justify-content: space-between;
div {
.stepCounterStep {
font-size: $h4-font-size;
color: $textPrimaryColor;
margin-right: 5px;
font-weight: 800;
}
.stepCounterStepOff {
font-weight: 500;
font-size: $h4-font-size;
color: $textAlternativeColor;
}
}
}
```

Лістинг colors

```
$primaryColor: #000000;
$secondaryColor: #111111;
$primaryBackground: #ffffff;
$disabledBackground: #f7f7f7;
$disabledColor: #d8d8d8;
$toxicGreen: #72ff49;
$yellowPrimary: #33cfff;
$yellowSecondary: #33cfff80;
$textPrimaryColor: #000000;
$textSecondaryColor: #d8d8d8;
$textAlternativeColor: rgb(185, 185, 185);
$textAlternativeColor1: rgba(186, 186, 186, 0.5);
$textPlatformsAndFeaturesColor: #949494;
$textButtonsWithBackground: #ffffff;
$borderPrimaryColor: #d8d8d8;
$secondaryBackgroundColor: #949494;
```

Лістинг fonts

```
@mixin font($font-family, $font-file) {
@font-face {
font-family: $font-family;
src: url($font-file + '.woff') format('woff');
font-weight: normal;
font-style: normal;
}
}
@include font('GothamPro', './fonts/GothamPro');
```

```
@include font('GothamPro-Bold', './fonts/GothamPro-Bold');
@include font('GothamPro-Medium', './fonts/GothamPro-Medium');
```

ЛІСТИНГ global

```
html {
  box-sizing: border-box;
  -ms-overflow-style: scrollbar;
  @media (min-width: 1120px) {
    height: 100%;
  }
}

*,
*::before,
*::after {
  box-sizing: inherit;
}

body {
  font-family: $font-default;
  position: relative;

  height: 100%;
  overflow-x: hidden;

  line-height: 1;

  opacity: 1;
  @media (min-width: 1120px) {
    display: flex;
    flex-direction: column;
    justify-content: center;
    flex: 1 1 auto;
  }
}

.applicationCalcBody {
  min-width: 320px;
}

h1,
h2,
h3,
h4,
h5,
h6 {
  margin: 0;

  font-family: $font-family-base;
}

h1,
.h1 {
  font-family: "GothamPro", GothamPro, sans-serif;

  font-size: $h1-font-size;
}

h2,
```

```

.h2 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: $h2-font-size;
}
h3,
.h3 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: $h3-font-size;
}
h4,
.h4 {
font-family: "GothamPro", GothamPro, sans-serif;

font-size: $h4-font-size;
}

blockquote {
margin-bottom: 0;
}

a {
font-family: $font-family-base;
font-weight: 600;
// font-size: 15px;
text-decoration: none;
letter-spacing: normal;

transition: color 0.3s ease;
&:hover {
// color: #1273e6;
text-decoration: none;
font-style: normal;
}
}

b,
strong {
color: #2c314e;
font-family: $font-family-base;
font-weight: 700;
}

#root {
margin: 0 auto;
width: fit-content;
@media (max-width: 1120px) {
width: 100%;
height: 100%;
}
}

```

Лістинг mixins

```

@mixin containerCreator {
width: 230px;

```

```

height: 110px;
border: 1px solid $textAlternativeColor1;
border-radius: 4px;
// opacity: 0.5;
display: flex;
justify-content: space-evenly;
h4 {
font-family: "GothamPro", GothamPro, sans-serif;

```

```

text-align: center;
margin: auto;
margin-left: 20px;
}
}
@mixin containerCreatorHover {
background-color: $disabledBackground;
}
@mixin roundContainerForIcon {
margin: auto;
margin-right: 20px;
width: 60px;
height: 60px;
border: solid 1px $disabledColor;
border-radius: 100%;
background-color: $primaryBackground;
}
@mixin icon {
background-position: center;
background-repeat: no-repeat;
}
@mixin contAnimation {
@keyframes animateDiv {
0% {
background-color: $primaryBackground;
}
50% {
background-color: $primaryBackground;
}
100% {
background-color: $disabledBackground;
}
}
animation: animateDiv ease-in 0.2s;
animation-fill-mode: forwards;
}

```

```

@mixin buttonElementAlign {
@keyframes moveElement {
0% {
transform: scaleX(0);
transform-origin: 0% 0%;
}
35% {
transform: scaleX(1);
transform-origin: 0% 100%;
}
70% {

```

```

transform: scaleX(1);
transform-origin: 100% 50%;
}
100% {
transform: scaleX(0);
transform-origin: 100% 50%;
}
}
animation: moveElement 1s;
animation-fill-mode: forwards;
}
@mixin buttonContentAlign {
@keyframes moveContent {
0% {
transform: scaleX(0);
transform-origin: 0% 0%;
}
99% {
transform: scaleX(1);
transform-origin: 0% 100%;
}
100% {
transform: scaleX(1);
transform-origin: 0% 100%;
box-shadow: 0 10px 35px -15px #33cfff80;
}
}
animation: moveContent 0.5s;
animation-fill-mode: forwards;
}

```

Лістинг theme

```

$font-default: GothamPro, sans-serif;
$font-family-headers: 'GothamPro-Medium', $font-default;
$font-family-base: 'GothamPro', $font-default;
$font-family-buttons: 'GothamPro-Medium', $font-default;

```

```

$font-size-base: 16px;

```

```

$h1-font-size: 30px;
$h2-font-size: 25px;
$h3-font-size: 20px;
$h4-font-size: 16px;
$h5-font-size: 14px;
$h6-font-size: 12px;

```

```

$small-text: 14px;

```

```

$h1-font-size-xs: 2rem;
$h2-font-size-xs: 1.625rem;
$h3-font-size-xs: 1.125rem;

```

Лістинг positions

```

@mixin absolute-center() {
position: absolute;
top: 50%;

```

```

left: 50%;
transform: translate(-50%, -50%);
}
@mixin absolute-all($num) {
position: absolute;
top: $num;
bottom: $num;
right: $num;
left: $num;
}

```

ЛІСТИНГ Aside

```

import React from 'react';
import { AsideContent } from './AsideContent';
export const Aside = (props) => {
const { toggle } = props;

return (
<div className="asideBarContainer">
<div className="quickViewButton" onClick={toggle}>
<div className="quickViewSideBarArrow"></div>
<p>Estimate</p>
</div>
<div
className="asideBarContent asideBarMainContainer"
style={{ width: '268px', height: '580px' }}
>
<AsideContent />
</div>
</div>
);
};

```

ЛІСТИНГ AsideContent

```

import React from 'react';
import { useAsideData } from './useAsideData';
export const AsideContent = (props) => {
const {
state,
descriptions,
dropDownDevTimeContent,
toggleNonDev,
toggleDev,
handleHideDescription,
handleShowDescription,
description,
isShowDescription,
isShowNonDev,
isShowDev,
} = useAsideData();
return (
<>
<div
className="asideBarContentMain"
style={{ backgroundColor: '#f7f7f7' }}
>

```

```

<div
  className="asideBarContent"
  onMouseOver={() => handleShowDescription(descriptions.summaryTime)}
  onMouseOut={handleHideDescription}
>
<h5 className="asideBarContentHeader">Summary time</h5>
<h2>{state.summaryTime}h</h2>
</div>
</div>
<div>
<div
  className="asideBarContentMain"
  onClick={toggleDev}
  onMouseOver={() => handleShowDescription(descriptions.devTime)}
  onMouseOut={handleHideDescription}
>
<div className="asideBarContent">
<h5 className="asideBarContentHeader">Development time</h5>
<h2>{state.devTime}h</h2>
</div>
{isShowDev && state.selectedSteps.length ? (
<div className="asideBarDropDownContent">
{dropDownDevTimeContent}
{state.devTime !== 0 ? (
<div className="asideBarDropDownContainer">
<p>Management time</p>
<p className="asideBarDropDownPrice">{state.managerTime}</p>
</div>
) : null}
</div>
) : null}
</div>
</div>
<div>
<div className="asideBarContentMain" onClick={toggleNonDev}>
<div
  className="asideBarContent"
  onMouseOver={() => handleShowDescription(descriptions.nonDevTime)}
  onMouseOut={handleHideDescription}
>
<h5 className="asideBarContentHeader">non-dev time</h5>
<h2>{state.nonDevTime}h</h2>
</div>
{isShowNonDev ? (
<div className="asideBarDropDownContent">
<div
  className="asideBarDropDownContainer"
  onMouseOver={() =>
handleShowDescription(descriptions.projectSetup)
}
  onMouseOut={handleHideDescription}
>
<p>Project setup</p>
<p className="asideBarDropDownPrice">
{(state.devTime !== 0
? state.projectSetupTime +
state.percentProjectSetup * (state.devTime / 100)

```

```

: 0
)
.toFixed(1)
.replace(/.0+$/, "")
</p>
</div>
<div
  className="asideBarDropDownContainer"
  onMouseOver={() =>
    handleShowDescription(descriptions.architecture)
  }
  onMouseOut={handleHideDescription}
>
<p>Architecture</p>
<p className="asideBarDropDownPrice">
  {(state.devTime !== 0
  ? state.architectureTime +
  state.percentArchitecture * (state.devTime / 100)
  : 0
  )
  .toFixed(1)
  .replace(/.0+$/, "")
  }
</p>
</div>
<div
  className="asideBarDropDownContainer"
  onMouseOver={() => handleShowDescription(descriptions.database)}
  onMouseOut={handleHideDescription}
>
<p>Database</p>
<p className="asideBarDropDownPrice">
  {(state.devTime !== 0
  ? state.databaseTime +
  state.percentDatabase * (state.devTime / 100)
  : 0
  )
  .toFixed(1)
  .replace(/.0+$/, "")
  }
</p>
</div>
<div
  className="asideBarDropDownContainer"
  onMouseOver={() => handleShowDescription(descriptions.network)}
  onMouseOut={handleHideDescription}
>
<p>Network</p>
<p className="asideBarDropDownPrice">
  {(state.devTime !== 0
  ? state.networkTime +
  state.percentNetwork * (state.devTime / 100)
  : 0
  )
  .toFixed(1)
  .replace(/.0+$/, "")
  }
</p>
</div>
<div

```

```

className="asideBarDropDownContainer"
onMouseOver={() => handleShowDescription(descriptions.swagger)}
onMouseOut={handleHideDescription}
>
<p>Swagger</p>
<p className="asideBarDropDownPrice">
{(state.devTime !== 0
? state.swaggerTime +
state.percentSwagger * (state.devTime / 100)
: 0
)
.toFixed(1)
.replace(/.0+$/, "")}
</p>
</div>
<div
className="asideBarDropDownContainer"
onMouseOver={() => handleShowDescription(descriptions.meetings)}
onMouseOut={handleHideDescription}
>
<p>Meetings</p>
<p className="asideBarDropDownPrice">
{(state.devTime !== 0
? state.meetingsTime +
state.percentMeetings * (state.devTime / 100)
: 0
)
.toFixed(1)
.replace(/.0+$/, "")}
</p>
</div>
<div
className="asideBarDropDownContainer"
onMouseOver={() =>
handleShowDescription(descriptions.continuousIntegration)
}
onMouseOut={handleHideDescription}
>
<p>Continuous Integration</p>
<p className="asideBarDropDownPrice">
{(state.devTime !== 0
? state.continuousIntegrationTime +
state.percentContinuousIntegration * (state.devTime / 100)
: 0
)
.toFixed(1)
.replace(/.0+$/, "")}
</p>
</div>
<div
className="asideBarDropDownContainer"
onMouseOver={() =>
handleShowDescription(descriptions.continuousDelivery)
}
onMouseOut={handleHideDescription}
>
<p>Continuous Delivery</p>

```

```

<p className="asideBarDropDownPrice">
  {(state.devTime !== 0
  ? state.continuousDeliveryTime +
  state.percentContinuousDelivery * (state.devTime / 100)
  : 0
  )
  .toFixed(1)
  .replace(/.0+$/, "")}
</p>
</div>
<div
  className="asideBarDropDownContainer"
  onMouseOver={() =>
  handleShowDescription(descriptions.releasePerSprint)
  }
  onMouseOut={handleHideDescription}
  >
  <p>Release per Sprint</p>
  <p className="asideBarDropDownPrice">
    {(state.devTime !== 0
    ? state.releasePerSprintTime +
    state.percentReleasePerSprint * (state.devTime / 100)
    : 0
    )
    .toFixed(1)
    .replace(/.0+$/, "")}
  </p>
</div>
<div
  className="asideBarDropDownContainer"
  onMouseOver={() =>
  handleShowDescription(descriptions.releasePerIntegration)
  }
  onMouseOut={handleHideDescription}
  >
  <p>Release per Integration</p>
  <p className="asideBarDropDownPrice">
    {(state.devTime !== 0
    ? state.releasePerIntegrationTime +
    state.percentReleasePerIntegration * (state.devTime / 100)
    : 0
    )
    .toFixed(1)
    .replace(/.0+$/, "")}
  </p>
</div>
</div>
) : null}
</div>
</div>
{isShowDescription ? (
  <div className="asideBarContentDescription">
    <p>{description}</p>
  </div>
) : null}
</>
);

```

```
};
```

ЛІСТИНГ StepsTime

```
import React, { useState, useEffect } from 'react';
import { useSelector } from 'react-redux';
import { calculateTime } from '../helpers/calculateFeaturesTime';

export const StepsTime = (props) => {
  const { id } = props;
  const state = useSelector((store) => store);
  const [featureTime, setFeatureTime] = useState(0);

  useEffect(() => {
    setFeatureTime(0);
    const step =
      state.selectedSteps.length &&
      state.selectedSteps.find((step) => step.id === id);
    step && state.selectedSteps.length
    ? step.features.map((feature) => {
      feature.isSelected
      ? setFeatureTime((prevState) => {
        return (
          parseFloat(prevState) +
          parseFloat(calculateTime(state, feature))
        );
      })
      : setFeatureTime((prevState) => parseFloat(prevState));
    })
    : setFeatureTime(0);
  }, [id, state]);

  return (
    <p style={{ cursor: 'pointer' }} className="asideBarDropDownPrice">
      {parseFloat(featureTime).toFixed(1).replace(/.0+$/, "")}
    </p>
  );
};
```

ЛІСТИНГ useAsideData

```
import React, { useState, useEffect } from "react";
import { useSelector, useDispatch } from "react-redux";
import { counter } from "../helpers/counter";
import { countTime } from "../store/actions";
import { useHistory } from "react-router-dom";
import { setStep, setDisplay } from "../store/actions";
import { StepsTime } from "./StepsTime";

export const useAsideData = () => {
  const state = useSelector((store) => store);
  const dispatch = useDispatch();
  const time = counter(state);
  const [isShowNonDev, setShowNonDev] = useState(false);
  const [isShowDev, setShowDev] = useState(false);
  const [description, setDescription] = useState("");
  const [isShowDescription, setShowDescription] = useState("");
  const history = useHistory();
```

```

const _handleMove = (item) => {
const step = `application-calculator/steps/step-${item.id}`;
history.push(step);
dispatch(setStep(item.id));
dispatch(setDisplay(true));
};
const handleShowDescription = (description) => {
setDescription(description);
setShowDescription(true);
};
const handleHideDescription = () => {
setShowDescription(false);
};
const toggleDev = () => {
setShowNonDev(false);
setShowDev(!isShowDev);
};
const toggleNonDev = () => {
setShowDev(false);
setShowNonDev(!isShowNonDev);
};

useEffect(() => {
if (
state.summaryTime !== time.summaryTime ||
state.devTime !== time.devTime ||
state.nonDevTime !== time.nonDevTime
) {
dispatch(countTime(time));
}
}, [dispatch, state.devTime, state.nonDevTime, state.summaryTime, time]);

const dropDownDevTimeContent = state.selectedSteps.map((item) => {
return (
<div
className="asideBarDropDownContainer"
onClick={() => _handleMove(item)}
key={item.title}
>
<p style={{ cursor: "pointer" }}>{item.title}</p>
<StepsTime id={item.id} />
</div>
);
});

const descriptions = {
summaryTime:
"Summing the non-development and development time, you receive the total time needed to deliver your product.",
devTime:
"It's the time required to implement given functionality. It includes time for business logic, UI (User Interface) and unit testing.",
nonDevTime:
"It's time required to set up the project, configure CI and CD, build database architecture, hold meetings and release the product.",
projectSetup:
"The time needed for initial project deployment and adjusting project properties and configuration settings",

```

swagger:

"Creation and maintenance of current data in the swagger, which is responsible for documenting app's APIs",

database:

"Work with project's database: database architecture, connections, etc.",

continuousIntegration:

"To maintain seamless quality assurance, developers integrate code into a shared repository several times a day",

meetings: "Time we devote to meetings with the development team",

architecture:

"App architecture building based on its functionality that has to be implemented, namely User Stories",

network:

"All tasks related to network: network monitoring, data parsing from xml, traffic management, etc.",

continuousDelivery:

"Development in short cycles which ensures automatic release of the part of the code that is ready for building and testing",

releasePerSprint:

"Preparation of the current version of the application to the demo, which is delivered once a sprint (usually every 2 weeks)",

releasePerIntegration:

"Release of the final version in production. The production server is updated, the application or site is deployed there",

};

return {

state,

descriptions,

dropDownDevTimeContent,

toggleNonDev,

toggleDev,

handleHideDescription,

handleShowDescription,

description,

isShowDescription,

isShowNonDev,

isShowDev,

};

};

ЛІСТИНГ BottomBar

```
import React from "react";
```

```
import { removeStep, setStep, setDisplay } from "../store/actions";
```

```
import { useSelector, useDispatch } from "react-redux";
```

```
import { useHistory } from "react-router-dom";
```

```
const BottomBar = () => {
```

```
  const store = useSelector((store) => store);
```

```
  const dispatch = useDispatch();
```

```
  const history = useHistory();
```

```
  const _handleRemove = (id) => {
```

```
    dispatch(removeStep(id));
```

```
    store.step === id + 1 &&
```

```
    store.availableSteps.length > store.step &&
```

```
    dispatch(setStep(store.step + 1));
```

```
    store.step === id + 1 &&
```

```
    store.availableSteps.length > store.step &&
```

```
    dispatch(setDisplay(false));
```

```
    store.step === id + 1 &&
```

```
    store.availableSteps.length > store.step &&
```

```

history.push(`/application-calculator/steps/step-${store.step + 1}`);

store.availableSteps.length === store.step &&
store.availableSteps.length === id + 1 &&
history.push(`/application-calculator/design`);
};
const _handlePlatform = () => {
history.push("/application-calculator/platform");
};
const _handleMove = (id) => {
const step = `application-calculator/steps/step-${id}`;
history.push(step);
dispatch(setStep(id));
dispatch(setDisplay(true));
};
const progress = (100 / store.availableSteps.length) * store.step;
const platformContentAlign = () => {
if (store.platform) {
return (
<div className="platformContent" onClick={() => _handlePlatform()}>
<h5 style={{ cursor: "pointer" }} title={"Move to platforms select"}>
{store.platform}
</h5>
</div>
);
}
};
const featuresContentAlign = () => {
if (!store.selectedSteps.length) {
return <h3 className="notSelectedFeatures">Not selected</h3>;
} else {
const features = store.selectedSteps.map((step, index) => {
return (
<div key={step.id} className="featuresBody">
<div
className="featuresRemove"
onClick={() => _handleRemove(step.id - 1)}
style={{ cursor: "pointer" }}
title={"Remove feature"}
></div>
<div
className="featuresContent"
onClick={() => _handleMove(step.id)}
>
<h5
style={{ cursor: "pointer" }}
title={`Move to ${step.title} step` }
>
{step.title}
</h5>
</div>
</div>
);
});
return features;
}
};

```

```

return (
  <div className="bottomBarContainer">
    <div className="progress-div">
      <div
        className="progress"
        style={{ width: `${progress}%`, transition: "1s" }}
      />
    </div>
    <div className="platform">
      <h4>Platform:</h4>
      {platformContentAlign()}
    </div>
    <div className="features">
      <h4>Features:</h4>
      <div className="featuresContainer">{featuresContentAlign()}</div>
    </div>
  </div>
);
};

```

```
export default BottomBar;
```

ЛІСТИНГ SaveAndContinue

```

import React from "react";
import { useDispatch, useSelector } from "react-redux";
import { passStep, setStep, setDisplay } from "../store/actions";
import { useHistory } from "react-router-dom";

const ButtonSubmit = (step) => {
  const dispatch = useDispatch();
  const history = useHistory();
  const state = useSelector((state) => state);
  const nextStepHolder =
    step < state.availableSteps.length
    ? `~/application-calculator/steps/step-${step + 1}`
    : "~/application-calculator/design";

  const _handleSubmit = (nextStep, step) => {
    step < state.availableSteps.length
    ? dispatch(setStep(step + 1))
    : dispatch(setStep(step));

    history.push(nextStep);
    dispatch(setDisplay(false));
  };

  return (
    <button
      className="saveAndContinueSubmit"
      type="submit"
      style={state.platform === "" ? { backgroundColor: "#f7f7f7" } : null}
      onClick={() => {
        _handleSubmit(nextStepHolder, step);
      }}
    >
    SAVE AND CONTINUE
  )

```

```

</button>
);
};
export default ButtonSubmit;

```

ЛІСТИНГ StepBack

```

import React from "react";
import { useHistory } from "react-router-dom";
import { useSelector, useDispatch } from "react-redux";
import { setTime, setStep, setDisplay } from "../store/actions";

const StepBack = (step) => {
  const history = useHistory();
  const store = useSelector((store) => store);

  const prevStep =
    history.location.pathname !== "/application-calculator/platform"
    ? step > 1
    ? `application-calculator/steps/step-${step - 1}`
    : "/application-calculator/platform"
    : "/application-calculator";
  const dispatch = useDispatch();
  const _handleSubmit = (prevStep, step) => {
    history.push(prevStep);
    dispatch(setStep(step));
    dispatch(setDisplay(false));
    store.selectedSteps.find((n) => {
      if (n.id === step - 1) {
        dispatch(setDisplay(true));
      }
    });
    step > 1 && dispatch(setTime(0));
  };

  return (
    <div
      className="stepBackButton"
      onClick={() => {
        _handleSubmit(prevStep, step);
      }}
    >
    <h2>#8249;</h2>
    <p>Back</p>
    </div>
  );
};
export default StepBack;

```

ЛІСТИНГ StepCounter

```

import React, { useState } from 'react';
import { useSelector } from 'react-redux';
import StepsQuickView from '../stepsQuickView';
import ClickAwayListener from '@material-ui/core/ClickAwayListener';

const StepCounter = (id) => {
  const [isShown, setIsShown] = useState(false);

```

```

const state = useSelector((state) => state);
const handleClickAway = () => {
  setIsShown(false);
};
return (
  <ClickAwayListener onClickAway={handleClickAway}>
    <div
      className="stepCounter"
      onClick={() => setIsShown(!isShown)}
      style={{ cursor: 'pointer' }}
      title={!isShown ? `Show available steps` : `Hide available steps`}
    >
      <div
        style={{
          width: '120px',
          display: 'flex',
          justifyContent: 'space-evenly',
        }}
      >
        <h4 className="stepCounterStep">STEP {id ? id : 0}</h4>
        <h4 className="stepCounterStepOff">
          OFF {state.availableSteps.length}
        </h4>
      </div>{' '}
      {isShown && <StepsQuickView />}
    </div>
  </ClickAwayListener>
);
};

```

```
export default StepCounter;
```

ЛІСТИНГ FeaturesContent

```

import React from "react";
import ButtonSubmit from "../Button/SaveAndContinueButton";
import { useDispatch, useSelector } from "react-redux";
import { selectFeature } from "../store/actions";
import { calculateTime } from "../helpers/calculateFeaturesTime";

export const FeaturesContent = (props) => {
  const dispatch = useDispatch();
  const store = useSelector((store) => store);
  const _handleCheck = (index, id, categoryId) => {
    dispatch(selectFeature(index, id, categoryId));
  };

  const _featuresMVPAlign = (id) => {
    return store.availableSteps[id].features.map((feature, index) => {
      const _editableCheck = () => {
        return (
          <label
            className="applicationCalcSelector"
            key={feature.title + index}
            title={
              feature.isEditable

```

```

? "Select feature"
: "This feature is must have"
}
>
<input
  className={
    feature.isEditable
    ? "applicationCalcSelectorCheckbox"
    : "applicationCalcSelectorCheckboxNotEditable"
  }
  type="checkbox"
  name="checkbox"
  checked={feature.isSelected}
  id={feature.id}
  onChange={() => {
    feature.isEditable && _handleCheck(index, id, store.categoryId);
  }}
/>
<div
  className={
    feature.isEditable
    ? "applicationCalcSelectorPseudo"
    : "applicationCalcSelectorPseudoNotEditable"
  }
></div>
<h5>{feature.title}</h5>
<h6>{calculateTime(store, feature).replace(/.0+$/, "")}</h6>
</label>
);
};
if (feature.isMvp) {
  return (
    <label
      key={`is-mvp-${feature.title}`}
      className="applicationCalcSelector"
    >
    { _editableCheck() }
    </label>
  );
}
});
};

const _featuresAdditionalAlign = (id) => {
  return store.availableSteps[id].features.map((feature, index) => {
    return (
      !feature.isMvp && (
        <label
          className="applicationCalcSelector"
          key={feature.title + index}
        >
        <input
          title={"Select feature"}
          className="applicationCalcSelectorCheckbox"
          type="checkbox"
          name="checkbox"
          checked={feature.isSelected}

```

```

id={feature.id}
onChange={() => {
  _handleCheck(index, id, store.categoryId);
}}
/>
<div className="applicationCalcSelectorPseudo"></div>
<h5>{feature.title}</h5>
<h6>{calculateTime(store, feature).replace(/.0+$/, "")}</h6>
</label>
)
);
});
};
const even = (element) => element !== false;

return (
<div>
<h4>Please chose the features bellow:</h4>
<div className="featuresSelectArea">
<div className="featureContent">
<div className="featureContentMVP">
<h4 style={{ margin: "10px 0" }}>Enough for MVP</h4>
{props.steps
? _featuresMVPAlign(props.id)
: props.history.push(props.nextStep)}
</div>
<div className="featureContentAdditionalFeatures">
{props.steps && _featuresAdditionalAlign(props.id).some(even) && (
<h4 style={{ margin: "10px 0" }}>Additional Features</h4>
)}
{props.steps
? _featuresAdditionalAlign(props.id)
: props.history.push(props.nextStep)}
</div>
</div>
</div>
<div className="featureContentSubmitContainer">
{ButtonSubmit(props.id + 1)}
</div>
</div>
);
};

```

Лістинг documnet

```

import React from "react";
import { useSelector } from "react-redux";

export const Document = () => {
  const store = useSelector((store) => store);
  const management = parseFloat(store.summaryTime * 7 * 0.15).toFixed(1);
  const backend = parseFloat(store.backendTime * 7).toFixed(1);
  const frontend = parseFloat(store.frontendTime * 7).toFixed(1);
  const design = parseFloat(store.designTime * 7 * 0.15).toFixed(1);
  const mobile = parseFloat(store.mobileTime * 7 * 0.15).toFixed(1);
  const total = management + backend + frontend + mobile + design;

```

```

const _featuresInfo =
store.selectedSteps &&
store.selectedSteps.map((item) => (
<>
<h4
style={{
fontWeight: "700",
lineHeight: "22px",
marginTop: "0",
marginBottom: "16px",
color: "#4a5566",
fontSize: "16px",
}}
>
{item.title}
</h4>
{item.features.map(
(item) =>
item.isSelected && (
<p
style={{
lineHeight: "22px",
color: "#8492a6",
fontSize: "16px",
marginTop: "8px",
}}
>
{item.title}
</p>
)
)}
</>
));
return (
<div
id="divToPrint"
style={{
height: "auto",
width: "auto",
backgroundColor: "white",
padding: "20px 48px",
}}
>
<p
style={{
lineHeight: "28px",
color: "#4a5566",
fontSize: "21px",
margin: "0",
width: "544px",
}}
>
Here is a rough <b>estimate</b> of your project!
</p>
<p
style={{
lineHeight: "28px",

```

```

color: "#4a5566",
fontSize: "21px",
margin: "0",
}}
>
{" "}
Thank you for finding out the cost ! We got the result and We think that
we can make an ideal estimate or even a <b>discount</b>.
</p>
<h4
style={{
fontWeight: "700",
lineHeight: "22px",
marginBottom: "16px",
color: "#4a5566",
fontSize: "16px",
marginTop: "32px",
}}
>
You pick this features:
</h4>
{ _featuresInfo }
<div>
<div
style={{
backgroundColor: "#e1e1ea",
height: "1px",
lineHeight: "1px",
margin: "12px 0px",
}}
>
&nbsp;
</div>
<div style={{ marginLeft: "224px" }}>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Design
</p>
<h3
style={{
margin: "auto 0",
lineHeight: "28px",
color: "#4a5566",

```

```

fontSize: "18px",
fontWeight: "400",
}}
>
&#36; {design}
</h3>
</div>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Mobile
</p>
<h3
style={{
margin: "auto 0",
lineHeight: "28px",
color: "#4a5566",
fontSize: "18px",
fontWeight: "400",
}}
>
&#36; {mobile}
</h3>
</div>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Backend
</p>
<h3
style={{

```

```

margin: "auto 0",
lineHeight: "28px",
color: "#4a5566",
fontSize: "18px",
fontWeight: "400",
}}
>
&#36; {backend}
</h3>
</div>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Frontend
</p>
<h3
style={{
margin: "auto 0",
lineHeight: "28px",
color: "#4a5566",
fontSize: "18px",
fontWeight: "400",
}}
>
&#36; {frontend}
</h3>
</div>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Management

```

```

</p>
<h3
style={{
margin: "auto 0",
lineHeight: "28px",
color: "#4a5566",
fontSize: "18px",
fontWeight: "400",
}}
>
&#36; {management}
</h3>
</div>
<div
style={{
display: "flex",
justifyContent: "space-between",
flexWrap: "wrap",
width: "272px",
}}
>
<p
style={{
lineHeight: "16px",
color: "#8492a6",
fontSize: "12px",
width: "72px",
}}
>
Total
</p>
<h3
style={{
margin: "auto 0",
fontWeight: "700",
lineHeight: "28px",
color: "#4a5566",
fontSize: "21px",
}}
>
&#36; {total} USD
</h3>
</div>
</div>
<div
style={{
backgroundColor: "#e1e1ea",
height: "1px",
lineHeight: "1px",
margin: "12px 0px",
}}
>
&nbsp;
</div>
</div>
<p
style={{

```

```

lineHeight: "16px",
marginTop: "0",
marginBottom: "16px",
color: "#8492a6",
fontSize: "12px",
}}
>
If you have any questions, reply to this email or contact us at{" "}
<a
href="mailto:hello@jetup.digital"
style={{
textDecoration: "none",
fontWeight: "700",
color: "#4a5566",
display: "inline-block",
}}
>
test@mail.com
</a>
.
</p>
<p
style={{
lineHeight: "16px",
marginTop: "0",
marginBottom: "16px",
color: "#8492a6",
fontSize: "12px",
}}
>
© 2020 Copyrights. All rights reserved.
</p>
<footer>
<div style={{ textAlign: "center" }}>
Page <span className="pageCounter"></span>/
<span className="totalPages"></span>
</div>
</footer>
</div>
);
};

```

Лістинг pdf

```

import React from "react";
import { jsPDF } from "jspdf";
import html2canvas from "html2canvas";
import { Document } from "./document";

export const PDF = () => {
  const printDocument = () => {
    const input = document.getElementById("divToPrint");

    html2canvas(input).then((canvas) => {
      const imgData = canvas.toDataURL("image/png");

      const pdf = new jsPDF();

```

```

pdf.addImage(imgData, "PNG", 0, 0);

const date = Date.now();

input.save(`CalculatorEstimate${date}.pdf`);
});
};

return (
<div className="m" style={{ height: "50px", overflow: "hidden" }}>
<div className="mb5" style={{ height: "50px" }}>
<button onClick={printDocument}>Print</button>
</div>
<Document />
</div>
);
};

```

ЛІСТИНГ StepHeader

```

import React from 'react';
import StepBack from '../Button/StepBack';
import StepCounter from '../components/Counter';

export const StepHeader = (header, id) => {
return (
<>
<div className="stepHeaderDiv">
{StepBack(id)}
{StepCounter(id)}
</div>
<div className="stepHeader">
<h2 className="stepHeader">{header}</h2>
</div>
</>
);
};

```

ЛІСТИНГ StepsQuickView

```

import React from "react";
import { useSelector, useDispatch } from "react-redux";
import { setStep, setDisplay } from "../store/actions";
import { useHistory } from "react-router-dom";

export const StepsQuickView = (ref) => {
const steps = useSelector((store) => store.availableSteps);
const store = useSelector((store) => store);
const dispatch = useDispatch();
const history = useHistory();

const _handleMove = (item) => {
if (item.isPassed) {
const step = `application-calculator/steps/step-${item.id}`;
history.push(step);
dispatch(setDisplay(false));
store.selectedSteps.find((n) => {

```

```

if (n.id === item.id) {
  dispatch(setDisplay(true));
}
});
dispatch(setStep(item.id));
}
};
const _disabledStyle = (item) =>
!item.isPassed
? { margin: "5px 15px", opacity: "0.5" }
: { margin: "5px 15px" };
const titles = steps
? steps.map((item) => (
<div
key={item.title}
style={_disabledStyle(item)}
onClick={() => _handleMove(item)}
>
<p
style={{
fontSize: "12px",
lineHeight: "18px",
}}
>
<b>{item.id}</b> {item.title} {" "}
</p>
<div
style={{
height: "1px",
backgroundColor: "#DFDFDF",
}}
/>
</div>
))
: "ERROR";
return (
<div
className="stepQuickView"
style={{
position: "absolute",
backgroundColor: "#F7F7F7",
width: "400px",
zIndex: "100",
marginTop: "20px",
marginLeft: "-280px",
maxHeight: "440px",
overflowY: "scroll",
}}
>
{titles}
</div>
);
};

```

Лістинг StepsEnum

```
export const RouterEnum = {
```

```
App: '/calculator',  
StepOne: '/step-1',  
StepTwo: '/step-2',  
StepThree: '/step-3',  
};
```

Додаток Б

Презентаційний матеріал

Кваліфікаційна робота бакалавра

Веб орієнтовна система оцінки собівартості товарів та послуг

Виконав: студент 4 курсу групи КН-17-2 Дасига В.В.
Керівник: к.т.н., доцент кафедри КНІТ Пасічник О. А.

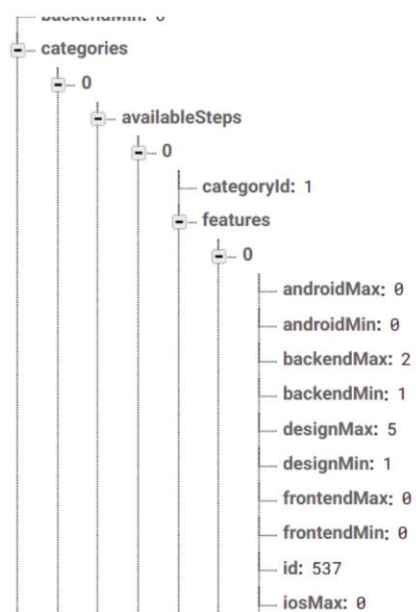
Актуальність

В наш час про використання ІТ, у тому числі новітніх, незалежно від сфери застосування пишуть і говорять багато.

Мета розробки кваліфікаційної роботи бакалавра, є веб застосунок для підрахунку собівартості товарів та послуг, який повинний забезпечити інформативність майбутніх клієнтів на етапі вирішення, який програмний продукт вони хочуть отримати.

Завдання

Мета розробки кваліфікаційної роботи бакалавра, є веб застосунок для підрахунку собівартості товарів та послуг, який повинний забезпечити інформативність майбутніх клієнтів на етапі вирішення, який програмний продукт вони хочуть отримати.



Вибір системи керування базами даних

База даних Firebase Realtime - це база даних, розташована в хмарі. Дані зберігаються у форматі JSON і синхронізуються в реальному часі з кожним підключеним клієнтом.

Висновок

В результаті проходження професійної практики розроблено веб застосунок для для підрахунку собівартості продукту. Для розробки використовувались найсучасніші технології, що призвело до високої швидкості виконання програми.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Дасига В. В. на захист дипломного проекту (роботи)
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Веб орієнтовна система оцінки собівартості товарів та послуг

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О. С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Дасига В. В. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки. повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 3,12 %, добре 21,88 %, задовільно 75,00 %.

шкалою ЄКТС: А 3,64 %, В 5,45 %, С 14,55 %, D 16,36 %, Е 60,00 %.

Методист факультету

(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент

Дасига В.В. виконав кваліфікаційну роботу бакалавра на тему "Веб орієнтовна система оцінки собівартості товарів та послуг" у відповідності до завдання в ньому обсязі

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

(підпис)

(прізвище та ініціали)

" 8 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Дасига В. В. допускається до захисту цього проекту

Завідувач кафедри

КНІТ

(назва)

Бармак О.В.

(підпис, прізвище, ініціали)

" 8 " 06 2021 р.

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 38.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 12%**

ID: 92869 Название: Веб орієнтовна система оцінки собівартості товарів та послуг Добавлено в БД: 2021-06-09 Авторы: В.В. Дасига Руководители: О.А. Пасічник Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	39082	377	17843 (46%)	176 (47%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы
90930	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-19 Авторы: Дасига В.В. Руководители: Скрипник Т.К. Консультанты: Опоненты:	14900 (38.0%)	139 (37.0%)

Ім'я користувача:
Кафедра КН

ID перевірки:
1008245430

Дата перевірки:
09.06.2021 13:44:03 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
09.06.2021 13:44:37 EEST

ID користувача:
100005671

Назва документа: PLAGIAT DASYHA (2)

Кількість сторінок: 54 Кількість слів: 5888 Кількість символів: 43531 Розмір файлу: 2.53 MB ID файлу: 1008317721

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

15.6%
Схожість

Найбільша схожість: 3.04% з Інтернет-джерелом (<https://www.sites.google.com/site/webtehnologijetawebdizajn/mova-ja>)

13.5% Джерела з Інтернету 355 Сторінка 56

6.11% Джерела з Бібліотеки 56 Сторінка 59

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 1

Підозріле форматування 15 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Веб орієнтовна система оцінки собівартості товарів та послуг

Автор: Дасига Владислав Вікторович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: Пасічник Олександр Анатолійович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

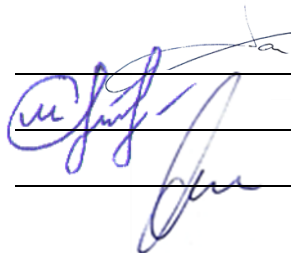
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 15,6% і адресується до ___ першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КНІТ



О.А. Пасічник

О.В. Мазурець

О. В. Бармак

Хмельницький національний університет
Кафедра комп'ютерних наук та інформаційних технологій

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента групи КН-17-2 Дасиги Владислава Вікторовича

за темою Веб орієнтовна система оцінки собівартості товарів та послуг

Актуальність і значення теми: система вирішує проблеми в оцінці вартості товарів та послуг, та надає можливість конфігурації фінального продукту, шляхом вибору запропонованих критеріїв.

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: алгоритми оптимізації процесу підрахунку собівартості, згідно обраних критеріїв використані доцільно, оскільки, досвід використання застосунок доволі зручний та зрозумілий і результат підрахунків містить всю необхідну для клієнта інформацію.

2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблена «Веб орієнтовна система оцінки собівартості товарів та послуг» використовується для оптимізації досвіду користувачів в процесі підрахунку собівартості.

3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі, веб орієнтовна система оцінки собівартості товарів та послуг, працює вірно, рекомендована оцінка «Добре».

Робота заслуговує на оцінку « добре »

Рецензент К.М.М. рол. Мезьєрчук І.К.