

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Демиденка Вадима Андрійовича
(ПІБ)

академічної групи 121-18-2
(шифр)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми Інженерія програмного забезпечення
(назва освітньої програми)

на тему: Створення веб-орієнтованого додатку з надання послуг
купівлі та здачі в оренду нерухомості на React.js

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтингово ю	інституційною	
кваліфікаційної роботи	доц. Приходченко С.Д.			
розділів:				
спеціальний	доц. Приходченко С.Д.			
економічний	доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	доц. Гуліна І.Г.			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних
систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » 2022 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 121-18-2

(група)

Демиденко В.А.

(прізвище та ініціали)

тема кваліфікаційної роботи

Створення веб-орієнтованого додатку

з надання послуг купівлі та здачі в оренду нерухомості на React.js

затверджена наказом ректора НТУ «ДП» від

№

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	13.05.2022 р.
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	27.05.2022 р.

Завдання видав

(підпис)

доц.Приходченко С.Д.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Демиденко В.А.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 21.06.2022

РЕФЕРАТ

Пояснювальна записка: 77 с., 33 рис., 3 дод., 20 джерел.

Об'єкт розробки: веб-орієнтований додаток з надання послуг купівлі та здачі в оренду нерухомості.

Мета кваліфікаційної роботи: створення веб-орієнтованого додатку з надання послуг купівлі та здачі в оренду нерухомості на React.js.

У вступі уточнюється сама постановка завдання, описується її головна мета, галузь безпосереднього застосування даної кваліфікаційної роботи і обґрунтовується актуальність теми.

У першому розділі досліджується предметна область та можливі існуючі рішення щодо виконання поставленого завдання, визначається актуальність завдання та описується призначення розробки, розроблюється постановка самого завдання.

У другому розділі виконується проектування і розробка самого програмного продукту, описується алгоритм і структура функціонування, обирається платформа для розробки і описується робота самої програми.

В економічному розділі описується трудомісткість самого розробленого програмного продукту, виконується підрахунок вартості роботи по створенню застосунку та проводиться розрахунок часу на його створення.

Практичне значення полягає у створенні програмного продукту та у покращенні практичних знань та отриманні навичок за час створення цієї роботи.

Актуальність програмного продукту визначається великою швидкістю розвитку сфери продаж та надання послуг, а також зробити ці процеси більш зручними та доступними для користувача.

Список ключових слів: ВЕБ, ВЕБ-ДОДАТОК, ПРОГРАМА, МАРКЕТПЛЕЙС, БРАУЗЕР.

ABSTRACT

Explanatory note: 77 p., 33 figs., 3 appx., 20 sources.

Object of development: web-oriented application for the provision of services for the purchase and rental of real estate.

The purpose of the qualification work: to create a web-based application for the purchase and rental of real estate on React.js.

The introduction specifies the statement of the task, describes its main purpose, the field of direct application of this qualification work and substantiates the relevance of the topic.

The first section examines the subject area and possible existing solutions for the task, determines the relevance of the task and describes the purpose of development, develops the formulation of the task.

The second section performs the design and development of the software product, describes the algorithm and structure of operation, selects a platform for development and describes the work of the program itself.

The economic section describes the complexity of the developed software product, calculates the cost of work to create an application and calculates the time to create it.

The practical significance lies in the creation of a software product and in the improvement of practical knowledge and skills during the creation of this work.

The relevance of the software product is determined by the high speed of development of sales and services, as well as make these processes more convenient and accessible to the user.

List of keywords: WEB, WEB APPLICATION, PROGRAM, MARKET PLACE, BROWSER.

ЗМІСТ

РЕФЕРАТ	Ошибка! Закладка не определена.
ABSTRACT	Ошибка! Закладка не определена.
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	Ошибка! Закладка не определена.
ВСТУП	Ошибка! Закладка не определена.
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	
1.1. Загальні відомості з предметної галузі	Ошибка! Закладка не определена.
1.2. Призначення розробки та галузь застосування.....	11
1.3. Підстави для розробки	11
1.4. Постановка завдання.....	12
1.5. Вимоги до програми або програмного виробу	12
1.5.1. Вимоги до функціональних характеристик.....	12
1.5.2. Вимоги до інформаційної безпеки	13
1.5.3. Вимоги до складу та параметрів технічних засобів	13
1.5.4. Вимоги до інформаційної та програмної сумісності.....	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	
2.1. Функціональне призначення програми.....	15
2.2. Опис застосованих математичних методів.....	15
2.3. Опис використаної архітектури та шаблонів проектування.....	16
2.4. Опис використаних технологій та мов програмування.....	18
2.5. Опис структури програми та алгоритмів її функціонування.....	22
2.6. Обґрунтування та організація вхідних та вихідних даних програми	25
2.7. Опис розробленого програмного продукту	26
2.7.1. Використані технічні засоби	26
2.7.2. Використані програмні засоби	26
2.7.3. Виклик та завантаження програми	27
2.7.4. Опис інтерфейсу користувача.....	28
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту	48
3.2. Розрахунок витрат на створення програми	51
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
Додаток А. Код програми.....	56
Додаток Б. Відгук керівника економічного розділу	76
Додаток В. Перелік файлів на диску.....	77

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

- БД – база даних;
- SEO – Пошукова оптимізація сайту
- JSON – JavaScript Object Notation, текстовий формат обміну даних
- ІС – інформаційна система;
- CSS – Cascading Style Sheets ;
- ОС – операційна система;
- ПК – персональний комп'ютер;
- ПЗ – програмне забезпечення;
- ІТ – інформаційні технології.

ВСТУП

Тема даної кваліфікаційної роботи є створення веб-орієнтованого додатку з надання послуг купівлі та здачі в оренду нерухомості, а також зробити його зручним для користування як з ПК, так і для користувачів телефонів.

Мета цієї кваліфікаційної роботи є вивчення предметної області розробки веб-сайтів, покращення та здобуття практичних навичок роботи з бібліотекою React.js, адаптувати веб-додаток для користувачів телефонів для збільшення відвідуваності сайту.

Ще якихось 10-15 років тому, щоб продати або купити квартиру, потрібно було публікувати оголошення в газеті або розклеювати листівки на дошках оголошення. І для вирішенні цієї проблеми приходить Інтернет. І для того, щоб стати власником бажаних квадратних метрів на допомогу приходять сайти з продажу нерухомості, на яких розміщуються оголошення як від приватних осіб, так і безпосередньо від самих забудовників. Завдяки цьому пошук необхідного житла стає значно простішим та зручнішим.

Сайт продажу нерухомості в першу чергу привабливий і зручний тим, що, не виходячи з дому, ви можете оцінити стан та зовнішній вигляд житла, оцінити зручність розташування нерухомості та багато всього іншого. Це дуже економить час, адже до ери Інтернету потрібно було об'їздити не один десяток об'єктів та витратити величезну купу часу перш ніж знайти бажаний.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості з предметної галузі

На сьогоднішній день ми спостерігаємо розвиток Інтернету що безумовно має великий вплив на багато сфер діяльності, як соціальну так і економічну. В соціальній сфері життя Інтернет дозволяє підтримувати спілкування та зв'язок з іншими людьми за допомогою соціальних мереж, а також отримувати різного роду необхідну інформацію. В економічній сфері розвитку набули інтернет-магазини та сервіси з надання послуг. Причинами успіху в економічній сфері для користувачів є зручність у використанні, більш детальна інформація про товар, можливість придбання з доставкою, можливість оцінити той чи інший товар або послугу. А для власників веб-сервісу це зручно тому що це дозволяє виводити свій бізнес на світовий ринок, що дає збільшення кількості відвідувань сервісу тим самим збільшувати кількість потенційних клієнтів.

Розвиток Інтернету також не оминув і проблему з пошуком нерухомості. На допомогу в цьому питанні приходять ті ж самі веб-сервіси, які дозволяють розміщувати та знаходити оголошення з продажу або здачі в оренду нерухомості. Вони значно спрощують як пошук необхідного житла так і розміщення оголошень від забудовників або від звичайних людей.

Раніше, 10-15 років тому, пошук необхідної нерухомості був надзвичайно довгим та дуже незручним. Для того щоб твоє оголошення хоч хтось побачив треба було розклеювати оголошення або давати оголошення в газеті тощо. Для того щоб зрозуміти чи відповідає оголошення твоїм потребам доводилося їхати і витратити велику кількість часу щоб побачити чи підходить воно чи ні. Також можна було зіткнутися з проблемою наявності документів, що також не спрощувало процес пошуку необхідної нерухомості.

Але тут є як свої переваги, так і свої проблеми.

Переваги:

– Отримання більш детальної інформації – на сьогоднішній день на сервісах з продажу нерухомості можна отримати детальну інформацію про необхідну нерухомість. Людина що розміщує оголошення також зацікавлена в наданні більш детальної інформації про нерухомість, тому що це також впливає на зацікавленість людей та викликає більше довіри до оголошення. Такою інформацією можуть бути велика кількість фото для того щоб отримати більш точне уявлення про нерухомість, план будівлі для ознайомлення з особливостями нерухомості, надання електронних копій документів власності, опис розташування поруч магазинів, АЗС, торговельних центрів, тощо.

– Збільшення цільової аудиторії – Інтернет надає змогу набагато більшій кількості людей побачити ваше оголошення, та збільшити шанси на знаходження потенційного клієнта.

– Зменшення кількості посередників – такого роду сервіси допомагають зв'язуватися з безпосередніми власниками або забудовниками без задіяння великої кількості посторонніх осіб.

Проблеми:

– гарантії інформаційної безпеки – збереження інформаційної безпеки стає все важчим так як зловмисники намагаються різними способами здобути персональні данні клієнтів. Тому на сьогоднішній день активно впроваджується сертифікація, авторизація та інші способи боротьби з шахрайством.

– незаконність або відсутність документів - на жаль не на всіх сервісах присутня можливість перевірити документи власності на нерухомість що призводить до великих проблем у майбутньому, або ж шахрайські дії у відношенні до продажу чужої нерухомості. Тому можна зіткнутися з тим що через відсутність або підробку документів шахраї можуть продати вам чужу нерухомість, можуть забрати гроші та зникнути тощо.

1.2. Призначення розробки та галузь застосування

Основною проблемою для більшості людей є пошук необхідної нерухомості яка буде відповідати їх вимогам і матеріальними можливостям. І для цього потрібно для клієнта, як забудовнику так і фізичній особі, надати якомога більшу кількість інформації та зробити це найбільш зручно та доступно для потенційного покупця.

На сьогоднішній день існують проекти які задовольняють більшість потреб своїх користувачів, але все ж таки не до кінця. Не всі подібні сервіси надають інформацію про ті об'єкти які знаходяться у розробці(будівництво або підготовка до продажу), а також надавати інформацію щодо документів та сертифікатів нерухомості.

Метою даної кваліфікаційної роботи є створення веб-сервісу який буде зручний і зрозумілий у використанні та буде надавати якомога більше інформації щодо нерухомості. І також для більшої зручності буде розділ на квартири і будинки.

1.3. Підстави для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма 121 «Інженерія програмного забезпечення»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 268-с від 18.05.2022 р;

завдання на кваліфікаційну роботу на тему “Створення веб-орієнтованого додатку з надання послуг купівлі та здачі в оренду нерухомості на React.js”.

1.4. Постановка завдання

Метою даної роботи буде створення веб-додатку який буде надавати користувачеві інформацію про нерухомість та надавати змогу для зв'язку з власником нерухомості для її придбання.

Даний веб-додаток буде створений на мультипарадигмній мові програмування JavaScript, з використанням бібліотеки React.js.

Для даної роботи було обрано бібліотеку React.js, яка оптимізує та спрощує розробку веб-додатку, а також добре підходить для розробки single page application(додаток у якого всі процеси відбуваються на одній сторінці). Але React має деяку проблематику у роботі з SEO, тому що пошукові боти в Інтернеті не можуть знайти данні на сторінці через те, що сторінка створюється за допомогою JS. Для того щоб прибрати даний недолік використовується Next.js.

У даній роботі для отримання даних з серверу буде використано REST API.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Кінцевий продукт матиме наступний функціонал:

- дозволяє користувачу звертатися до власників нерухомості для її покупки заповнивши необхідні текстові поля(Name, Phone Number, Email та same Message).
- перегляд всіх доступних пропозицій;
- перегляд деталей обраної нерухомості;
- переглядати план будівлі та документи пов'язані з нею;
- завантажувати план будівлі;
- переглядати статус нерухомості(продається, у розробці).

Веб-застосунок повинен надавати необхідну користувачам інформацію в зручному вигляді, мати зрозумілий та зручний інтерфейс, а також сприяти взаємодії продавця з клієнтом.

1.5.2. Вимоги до інформаційної безпеки

Основні вимоги до інформаційної безпеки:

- конфіденційність інформації;
- цілісність даних;
- доступність інформації;
- автентичність даних;

Веб-додаток повинен забезпечувати надійний захист даних користувача та їх верифікацію.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для роботи з веб-додатком необхідний браузер. Характеристики необхідні для роботи більшості браузерів для операційної системи Windows:

- ЦП: Pentium 4;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD/ATI;
- відеопам'ять: 128 МБ;
- накопичувач: 150 Гб;
- оперативна пам'ять: 2 Гб.

Для мобільних платформ необхідні мінімальні версії операційних систем:

- iOS 11.0 і вище;
- Android 4.4 і вище.

1.5.4. Вимоги до інформаційної та програмної сумісності

Старі версії браузерів можуть не підтримувати нововведення, бібліотеки

або фреймворки. В Сучасних браузерях є функція автоматичного оновлення, тому проблем у користувачів не повинні виникати при роботі в найпоширеніших браузерях.

Для підтримки були обрані наступні браузери:

- Chrome від 87 версії;
- Opera від 74 версії;
- Edge від 88 версії;
- Safari від 13.1 версії;
- IE від 11 версії;
- Android від 90 версії;
- Baidu від 7.12 версії;
- Firefox від 78 версії;
- iOS Safari від 13.4версії;
- Opera mobile від 62 версії;
- Samsung від 13.0 версії.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

В результаті виконання даної кваліфікаційної роботи повинен бути створений додаток який має бути сумісним для роботи в різноманітних середовищах, в результаті чого додаток буде запускатися в сучасних браузерах і не буде залежати від операційної системи та від того який пристрій ви використовуєте, чи то ПК або телефон.

Кінцевий продукт матиме наступний функціонал:

- дозволяє користувачу звертатися до власників нерухомості для її покупки заповнивши необхідні текстові поля(Name, Phone Number, Email та same Message).
- перегляд всіх доступних пропозицій;
- перегляд деталей обраної нерухомості;
- переглядати план будівлі та документи пов'язані з нею;
- завантажувати план будівлі;
- переглядати статус нерухомості(продається, у розробці).

Веб-застосунок повинен надавати необхідну користувачам інформацію в зручному вигляді, мати зрозумілий та зручний інтерфейс, а також сприяти взаємодії продавця з клієнтом який використовує ПК так і з клієнтом що використовує телефон.

2.2. Опис застосованих математичних методів

Під час проектування та розробки даної інформаційної системи використовувалися лише прості арифметичні дії. Математичні методи в даній кваліфікаційній роботі не використовувалися.

2.3. Опис використаної архітектури та шаблонів проектування

В ході виконання даної кваліфікаційної роботи, для обміну даними між сервером та клієнтом, було обрано архітектурний стиль REST.

REST – це спосіб який дає змогу взаємодіяти і обмінюватися даними сервера, своєрідна оболонка взаємодії компонентів розподіленого додатку в мережі. REST є узгодженим набором обмежень, який потрібно враховувати під час проектування розподіленої системи. У випадку даної кваліфікаційної роботи цей підхід призведе до підвищення продуктивності і спрощення архітектури. У широкому сенсі компоненти в REST взаємодіють на зразок взаємодії клієнтів і серверів. Єдиного стандарту якого всі дотримуються у неї немає, тому, що це не протокол, а саме цілий архітектурний стиль. Цим вона і відрізняється від багатьох аналогічних.

Більшість великих компаній розробляють для внутрішнього використання або для своїх клієнтів цей інтерфейс, адже він здатний забезпечити спілкування між двома системами і надавати зручну для обробки інформацію.

В мережі Інтернет виклик віддаленої процедури може являти собою звичайний HTTP-запит, зазвичай використовується GET або POST. І тому такого роду запит називають «REST-запит», а необхідні дані передаються в якості параметрів запиту.

Для веб-служб, побудованих з використанням REST, застосовують термін «RESTful». Більшість RESTful-реалізацій використовують такі стандарти, як HTTP, URL, JSON і рідше, XML.

Переваги використання даної архітектури:

- компоненти систем взаємодіють в більшому масштабі;
- всі інтерфейси мають загальний характер;
- частини можна впроваджувати незалежно одна від одної;
- є проміжні елементи, які посилюють безпеку з'єднання.

Недоліки використання даної архітектури:

- надлишкова вибірка - коли запит на кінцеву точку API надає інформацію, яка не вимагається клієнтами, або ними не використовується.
- недостатня вибірка - можливі ситуації, коли клієнт може отримувати по декілька запитів, щоб отримати всю інформацію, яка йому потрібна.

Фактичні маніпуляції з DOM та форматування виводу абстрагуються в Директиви та Фільтри.

Next.JS – це інструмент, який використовується в основному для створення веб-сайтів, які відображаються на стороні сервера, які динамічно генерують HTML через сервер під час кожного випадку отримання нового запиту. В даній кваліфікаційній роботі використовуються такі методи NEXT.JS як `getStaticProps`, `getStaticPaths`, `getServerSideProps`.

Основні моментами в використанні методу `GetStaticProps`:

- Цей метод в основному використовується для отримання даних сторінок під час збірки проекту.
- Статичні дані цих сторінок оновлюються тільки у тому випадку коли проект заново проводить збірку проекту.
- Перевага використання `GetStaticProps` полягає в тому, що він дозволяє статично генерувати всі сторінки. В результаті отримання всіх статичних сторінок з усіх доступних даних `GetStaticProps` генерує найшвидший час завантаження.
- Оскільки сторінки заповнюються змістом на стороні серверу клієнт вже отримує повноцінну сторінку з даними та пошукові боти браузерів можуть просканувати зміст сторінок.

`getStaticPaths` є одним з методів генерації статичного сайту (SSG), який є унікальним для Next.js поряд з `getStaticProps`. `getStaticPaths` використовується для динамічних маршрутів, за допомогою якого ми визначаємо список шляхів, які мають бути відтворені в HTML під час створення.

`getServerSideProps` – це метод який відповідає за рендеринг елементів веб-додатку зі сторони серверу. Спочатку відбувається заповнення необхідних

даних на стороні сервера, а потім йде процес відправлення готової сторінки на клієнтську частину додатку.

2.4. Опис використаних технологій та мов програмування

Дана кваліфікаційна робота була виконана з використанням даних технологій:

- React.js;
- Next.js;
- Node.js;
- TypeScript;
- Modules.css;
- JavaScript;
- Axios;
- SCSS.

React - це JavaScript бібліотека яка була створена компанією Facebook для розробки інтерфейсу користувача (UI, user interface). В React використовується JSX синтаксис який робить розробку UI більш зручною. Особливість бібліотеки React є віртуальний DOM, за рахунок цього пришвидшується робота з DOM браузера. Це відбувається за рахунок того, що React створює віртуальний об'єкт DOM-дерева в якому він змінює данні та при необхідності він вносить зміни у реальний DOM браузера. Завдяки цьому браузеру не потрібно заново створювати нову сторінку в DOM-дереві тим самим економляться ресурси браузера таким чином пришвидшуючи роботу.

JavaScript – це мультипарадигменна мова програмування яка може підтримувати функціональний, імперативний, а також об'єктно-орієнтований стилі програмування. Початковою метою використання JavaScript було надання можливості користувачу взаємодіяти з інтерфейсом веб-сайту. Але згодом JavaScript отримав визнання серед розробників і це призвело до розширення сфер застосування цієї мови програмування. І саме тому, на

сьогоднішній день, JavaScript можна зустріти не тільки в веб-сайтах, а й в ПО, мобільних додатках та різного роду приладах.

В даному проекті в роботі з JavaScript використовується ES2017 стандарт, але так як сучасні браузер не підтримують новий ES стандарт тому в роботі над проектом використовують webpack та babel який перетворює код відповідно до стандарту ES6.

Характерними змінами в ES6 відносно ES5 стали: константи, блочні змінні та функції, стрілочні функції, параметри функції за замовчуванням, використання spread оператора (...args), використання template літерала (``) у роботі зі рядками, геттери та сетери.

TypeScript — це надмножина JavaScript, яка насамперед надає необов'язкові статичні типи, класи та інтерфейси. Однією з головних переваг є можливість IDE(інтегроване середовище розробки) надавати багатше середовище для виявлення поширених помилок під час введення коду. Розробниками TypeScript є Microsoft. Тому ми можемо спостерігати схожий з C# синтаксис.

В великому проекті на JavaScript застосування TypeScript може призвести до збільшення надійності програмного забезпечення, але його можна буде розгорнути там, де буде працювати звичайна програма яка написана на JavaScript.

Modules.css. Модульний CSS застосовується для нових компонентів системи проектування тому що він добре працює з існуючими глобальними стилями та елементами інтерфейсу.

Модулі CSS чудово підходять для розробки, оскільки вони допомагають писати багаторазові компоненти з ізольованими стилями. Відповідно, файли CSS, у яких усі назви класів та назви анімації за замовчуванням мають локальну область. З модулями CSS класи CSS мають посилатися у файлі JavaScript через явне прив'язування до файлу стилів. Спочатку компілятор оновить файл CSS на етапі збірки, замінивши клас селектора CSS, на який

посилається в розмітці, унікальним набором символів. А далі файл JavaScript буде оновлено шляхом заміни класу CSS на новий вбудований рядок.

Цей підхід призначений для вирішення проблеми глобальної області видимості в CSS. Розробники можуть називати свої селектори CSS будь-яким чином, не турбуючись про ненавмисні наслідки в інших областях коду. Створення модуля CSS зрештою дуже схоже на створення типового файлу CSS.

Node.js - являє собою платформу з відкритим кодом для виконання високо-продуктивних мережевих застосунків, які написані на мові програмування JavaScript.

Модульність Node.js надає можливості створювати невеликі додатки без необхідності підтримки величезної інфраструктури, тому як більша частина функціоналу в результаті не використовується. При розробці додатку на Node.js розробник сам може вирішувати, що йому потрібно, і, за необхідності, розширити проект необхідними для нього бібліотеками або іншими ресурсами.

Окрім вбудованого функціоналу, який постачає сам Node.js, в його склад входить також менеджер модулів npm (node package manager). Його основна ідея – це створення маленького програмного блоку, який може вирішувати одну конкретну проблему, але робить це швидко та якісно. Це надає нам можливість розбивати на блоки власні великі проекти з цих маленьких побудованих блоків. Можна додати що, цей підхід задовольняє принцип відкритості та багаторазового використання коду.

Next.js — це інтерфейсний фреймворк React з відкритим вихідним кодом, який надає додаткові можливості оптимізації, як-от рендеринг на стороні сервера (SSR) і створення статичного сайту. Next.js базується на бібліотеці React, тобто програми Next.js користуються перевагами React і просто додають додаткові функції.

Рендеринг на стороні сервера. SSR дозволяє серверу отримати доступ до всіх необхідних даних і разом обробити JavaScript для відтворення сторінки.

Потім сторінка повністю відправляється назад у браузер і негайно відтворюється. SSR дозволяє веб-сторінкам завантажуватися за часткову частину часу та покращує роботу користувачів із додатковою швидкістю реагування.

Пошукова оптимізація (SEO). Використання SSR також дає вам перевагу в SEO, що допомагає вашому сайту відобразитися вище на сторінках результатів пошуку. SSR покращує рейтинг веб-сайтів для SEO, оскільки вони завантажуються швидше, а більшу частину вмісту сайту можна сканувати за допомогою SEO-трекерів. Тег `<head>`. Next.js також дозволяє редагувати тег `<head>` сайту, чого ви не можете зробити в React. Сам же тег `<head>` є основною частиною метаданих веб-сторінки і вносить свій внесок у рейтинг SEO сайту.

Загалом, Next.js вважається більш повнофункціональною версією React, яка підтримує інтуїтивність, якою відомий React Framework.

Axios – являє собою клієнтську бібліотеку HTTP. Вона використовує проміси за замовчуванням і може працювати як на клієнті, так і на сервері, що безумовно робить її придатною для отримання даних під час рендерингу зі сторони сервера. Оскільки вона використовує проміси значить її можна поєднати з методами `async/await`, для того щоб отримати стислий і простий підхід у використанні API.

Sass - це метамова на основі CSS, призначення якої збільшення рівня абстракції CSS-коду і спрощення роботи з файлами каскадних таблиць стилів.

Синтаксис цієї мови доволі гнучкий, він також пропонує вирішення безлічі проблем, які присутні при застосуванні звичайного CSS коду. Дана мова надає можливості додавати певну логіку (`@if`, `each`), математичні дії (можна складати як числа, рядки, так і кольори). Важливим також є те що з'являється можливість вкладеного стилізування компонентів, що насамперед збільшує швидкість отримання потрібного або очікуваного результату від заданих стилів, шляхом значного прискорення пошуку усіх потрібних стилів, що впливають на певний елемент.

2.5. Опис структури програми та алгоритмів її функціонування

Для більш кращого представлення алгоритмів роботи даного веб-додатку було використана UML діаграма, на якій представлено взаємодію користувача з інтерфейсом, а також з власником тої чи іншої нерухомості. Виходячи з діаграми 2.1 ми бачимо що взаємодія користувача відбувається за допомогою форм, які користувач має заповнити своїми даними. І тим самим дає можливість власнику зв'язатися безпосередньо з користувачем.

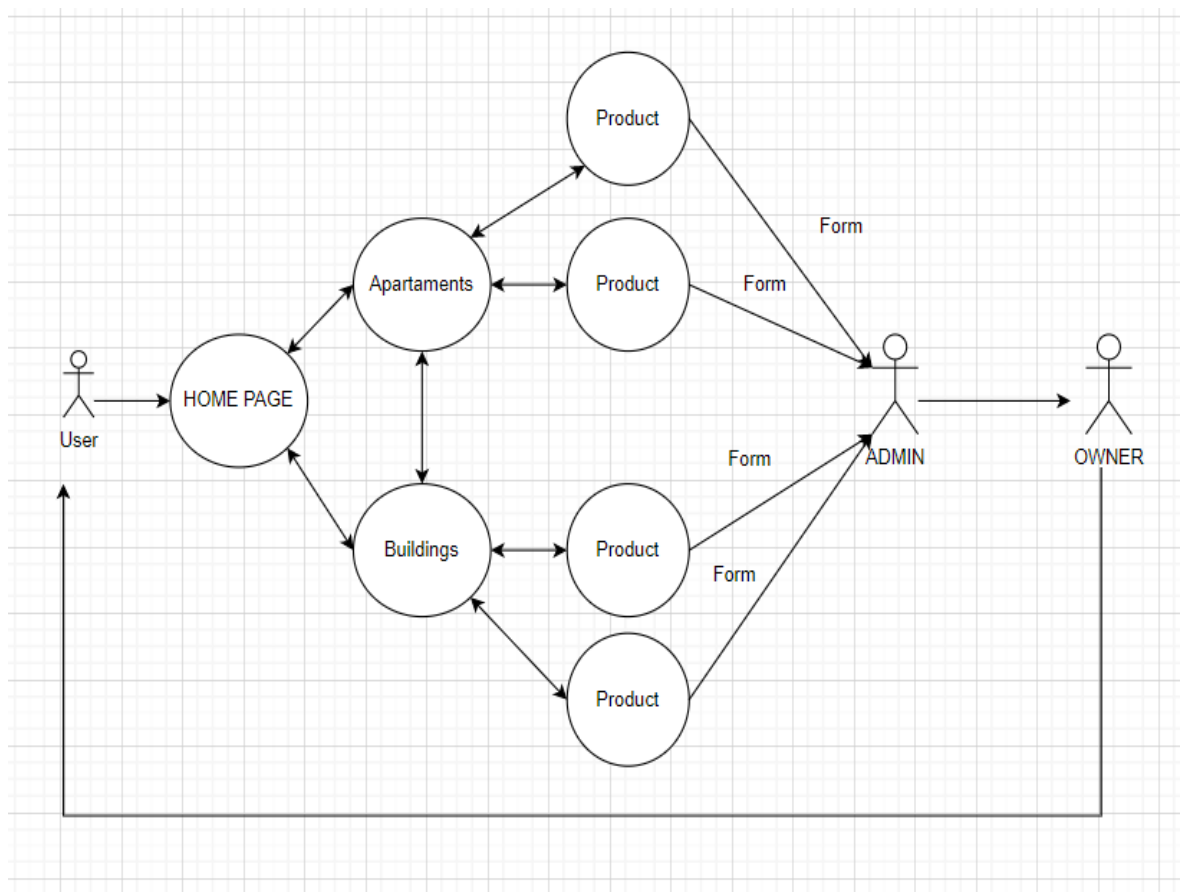


Рис. 2.1. – Діаграма взаємодії користувача з веб-додатком.

Даний веб-додаток створений з застосуванням компонентного підходу. Цей підхід є однією з особливостей розробки на React. Компонентний підхід має на увазі те що кожен елемент на сторінці виступає окремим компонентом, це дає змогу перевикористовуватися в різних частинах проекту. І при цьому компоненти не залежать один від одного.

Використання компонентного підходу в React не те що рекомендується, а навіть пропагується використання даного підходу у самій же документації. Так як цей підхід дуже зручний у плані розробки. Цей підхід масштабується, тому компоненти в проекті можуть бути як невеличкими та незначними, так і великими секціями, блоками або навіть окремими сторінками.

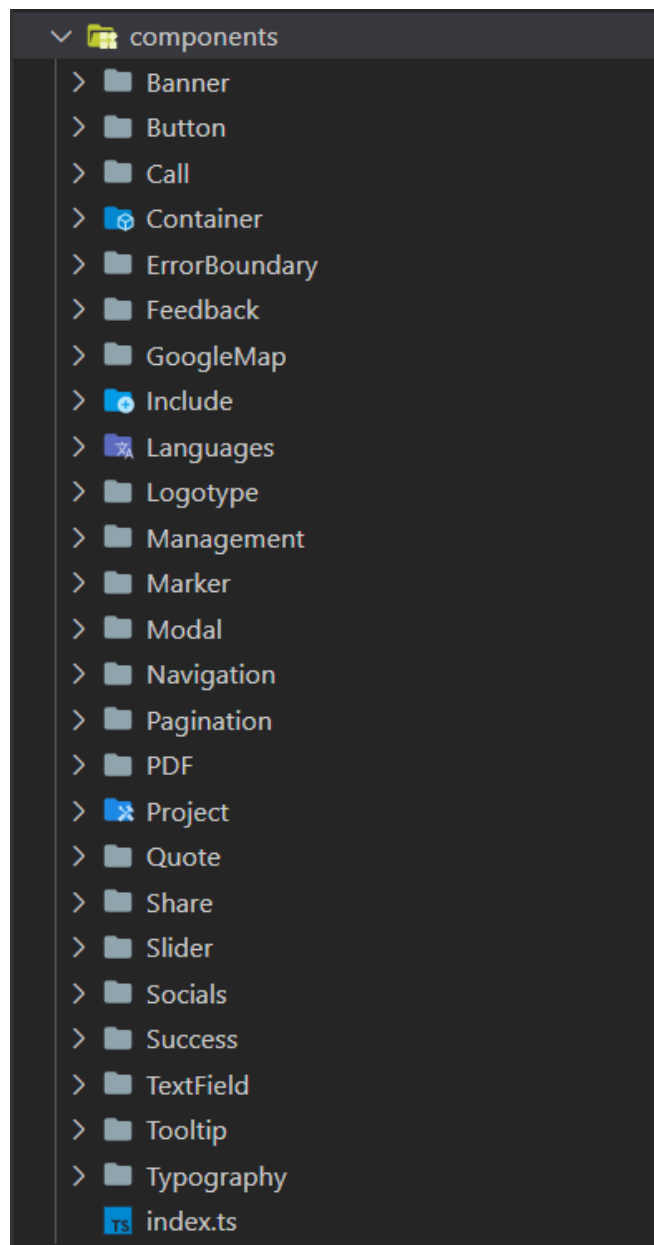


Рис. 2.2. – Директорія компонентів проекту.

Загальна структура файлів проекту:

- `node_modules` – директорія де зберігаються npm модулі, які використовувалися в ході розробки веб-додатку;
- `public` – директорія в якій знаходяться всі іконки та зображення які використовувалися під час розробки;
- `pageComponents` – директорія в якій знаходяться всі сторінки проекту.
- `pages` – в цій директорії зберігаються сторінки, а також здійснюється маршрутизація проекту, а також відбувається статична генерація сторінок або серверна генерація сторінок.
- `services` – в даній директорії знаходяться файли взаємодії з сервером.
- `styles` – тут знаходяться глобальні стилі проекту, підключення шрифтів, а також CSS-змінні.
- `types` – в цій директорії знаходиться типізація даних компонентів проекту.
- `next.config.js` – в дном файлі йде налаштування фреймворку `next.js`.
- `images.d.ts` – в цьому файлі відбувається типізація зображень та іконок використаних в даному проекті.

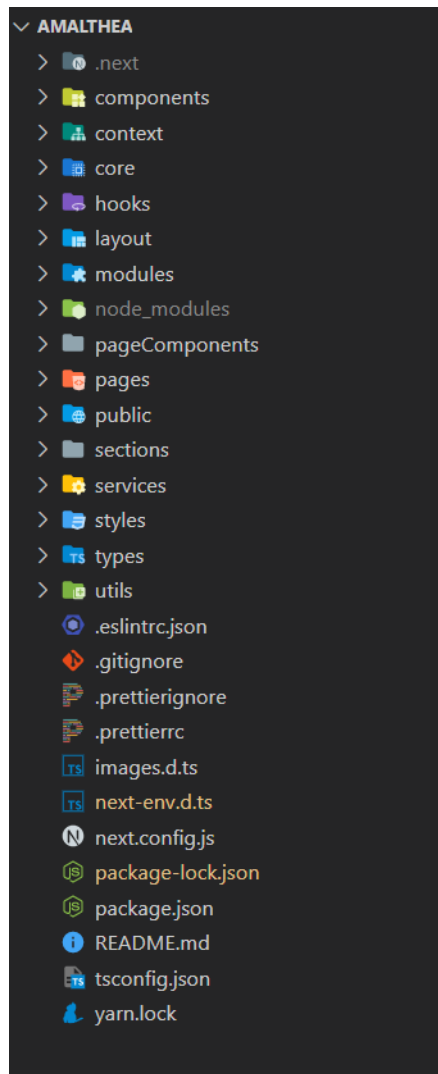


Рис. 2.3. – Перелік файлів проекту

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Обмін даних між сервером та клієнтським застосунком відбувається у форматі JSON. Так як цей тип даних зручний як для обробки на сервері, так і для роботи на клієнті частині додатку.

Вхідними даними до цього проекту можуть бути різного роду дані. Сервіс приймає дані користувача такі як ім'я, номер телефону, адресу електронної пошти, а також фільтрує ті проекти що маються на квартири або будівлі.

Вихідні дані які сервіс надає користувачеві інформацію про забудовників, геодані та місце розташування будівель, надання інформації про те продається

об'єкт або здається в оренду, чи добудована будівля або знаходиться у стані будівництва. Також якщо зайти на саму сторінку проекту можна отримати дані про знаходження поруч магазинів, кількість місць на парковці, дату будівництва, наявність ліфту у будівлі, кількість поверхів, файл з документами будівлі у форматі .pdf, тип будівлі.

2.7. Опис роботи розробленого програмного продукту

2.7.1. Використані технічні засоби

Користувач перш ніж скористатися даним веб-додатком повинен мати систему, яка зможе запускатися та працювати з сучасними браузерами, тому користувачеві необхідно мати такі мінімальні параметри ЕОМ які рекомендують розробники більшості браузерів:

- ЦП: Pentium 4.
- Відеоадаптер: 3D адаптер nVidia, Intel, AMD/ATI.
- Відеопам'ять: 128 МБ.
- Накопичувач: 150 Гб.
- Оперативна пам'ять: 2 ГБ.

2.7.2. Використані програмні засоби

В ході розробки проекту було використано такі програмні засоби як:

- VS Code
- Node.js
- Git

Visual Studio Code - це доволі легкий, але потужний редактор коду, який доступний для усіх популярних на сьогоднішній день операційних систем таких як Windows, macOS та Linux. Він має вбудовану підтримку JavaScript, TypeScript та Node.js. Також в даному редакторі коду є дуже розвинута

екосистема розширень які допомагають у роботі з JavaScript, Node.js а також TypeScript.

Підтримує виділення синтаксису а також автозаповнення за допомогою IntelliSense, який забезпечує розумні доповнення на основі типів та змінних, визначень функцій і модулів які імпортуються.

Node.js — це платформа яка призначена саме для виконання високопродуктивних мережевих застосунків, які були написані мовою програмування JavaScript. Платформа в основному використовується для створення серверних програм.

В Node.js використовується розроблений компанією Google двигун V8. Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в не блокуючому режимі та визначенні обробників зворотніх викликів (callback).

Git представляє собою систему управління розподіленими версіями з відкритим вихідним кодом.

Система контролю версій дозволяє завантажувати код, який зберігається в Git, а також постійно змінюється в міру додавання коду. Крім того, багато розробників можуть додавати код паралельно. Таким чином, система контролю версій допомагає впоратися з цим, зберігаючи історію змін, що відбулися. Крім того, Git надає такі функції, як розгалуження та злиття Git має віддалений репозиторій, який зберігається на сервері, і локальний репозиторій, який зберігається на комп'ютері кожного з розробників. Це означає, що код не просто зберігається на центральному сервері, а повна копія коду є на всіх комп'ютерах розробників.

2.7.3. Виклик та завантаження програми

Даний веб-застосунок завантажується після переходу по URL - стандартизованій адресі за якою додаток буде доступний в мережі інтернет,

за допомогою одного браузеру. Прикладом на основних десктопних ОС, таких як Windows 10, Linux, Mac OS, є такі браузери, як Google Chrome, Firefox, Brave, Microsoft Edge. Для мобільних платформ підтримуються браузери Google Chrome, Dolphin, Opera Mobile, Mozilla Firefox, Safari, Microsoft Edge.

Додаток гарантовано запускається на останніх двох версіях усіх браузерів які офіційно підтримуються розробниками, на практиці додаток працює і на старіших версіях та гарантій для їх успішної роботи немає.

2.7.4. Опис інтерфейсу користувача

Після того як ми запускаємо веб-додаток користувач потрапляє на головну сторінку, яка має назву Home. Це одна з основних сторінок в ієрархії проекту, на ній ми можемо взаємодіяти з елементами меню які допомагають виконувати навігацію по веб-додатку. Даний веб-додаток орієнтований на німецького користувача, але для більшої доступності для всіх користувачів в інтерфейсі веб-додатку використовується англійська мова.

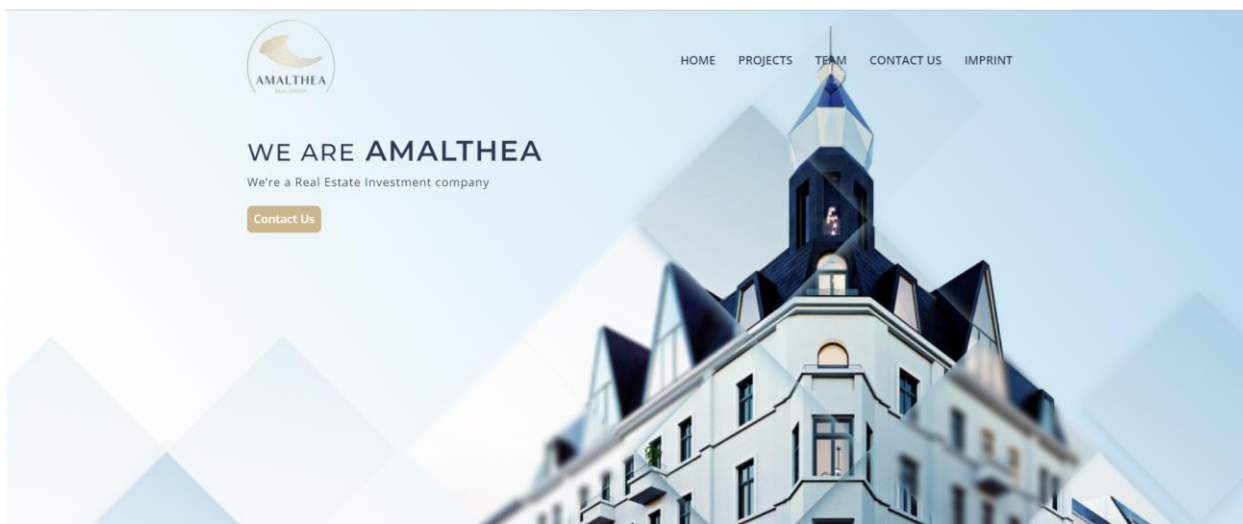


Рис. 2.4. – Головна сторінка з меню навігації



Рис. 2.5. – Головна сторінка з меню навігації на iPhone 12 Pro

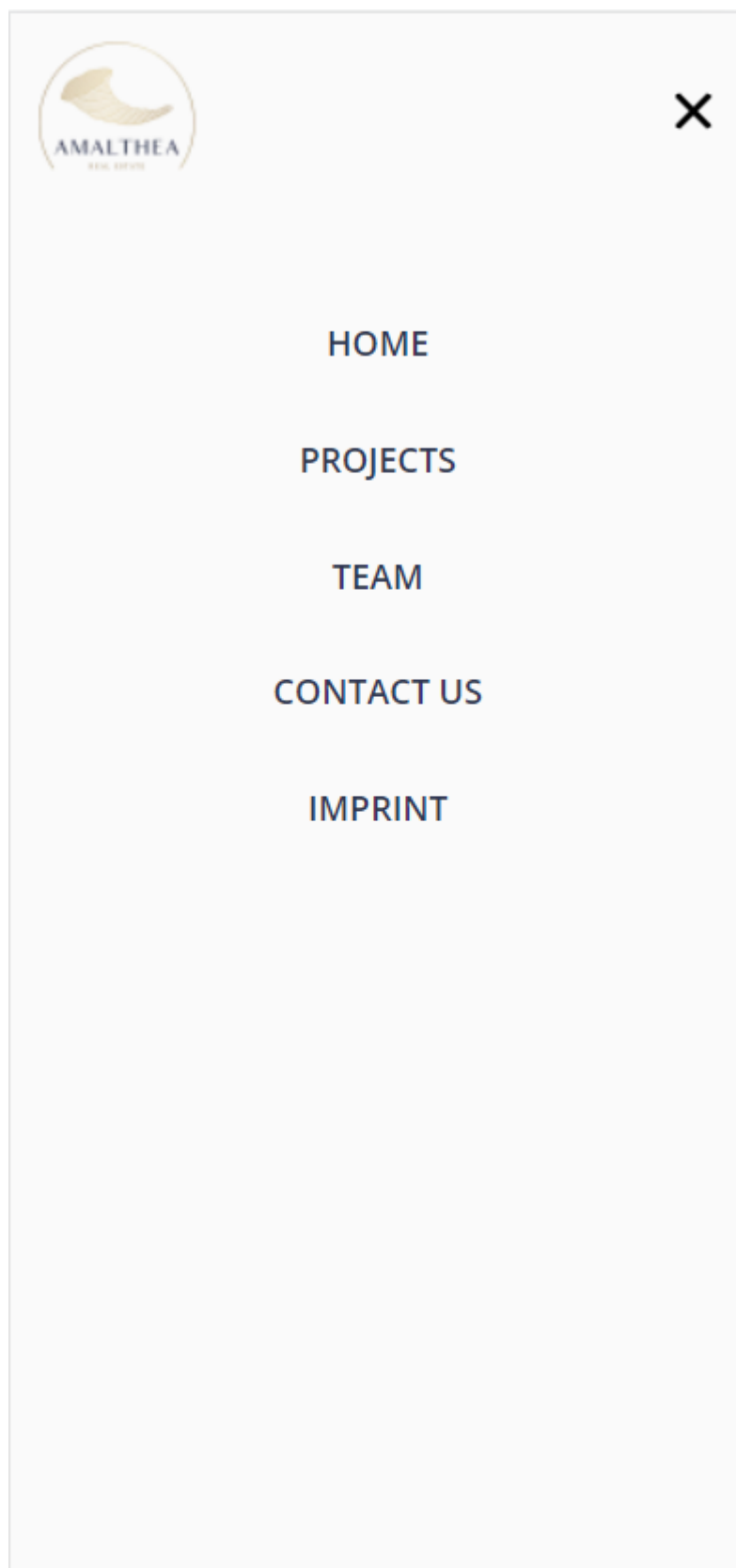


Рис. 2.6. – Розгорнуте меню навігації на iPhone 12 Pro

Коли користувач заходить на сторінку, його зустрічає свого роду слайдер який змінює своє фонове зображення, в шапці сторінки ми бачимо навігаційне меню, яке допомагає нам швидше та зручніше користуватися веб-додатком(рис. 2.4). Також ви можете бачити як цей слайдер та навігаційне меню працюватиме на мобільних пристроях(рис.2.5., рис.2.6.), для наглядного прикладу було обрано iPhone 12 Pro.

Our Business Philosophy & Investment Criteria


For more than 15 years we have been active in the field of real estate investment, project development, and VC investments. Here, with an experienced and dynamic team, we pursue time tested investment cases of strictly grade B+/A+ properties with significant value-add potential in major cities in Germany and Portugal, as well as synergetic investments into related Prop-Tech companies.

We are experienced real estate and risk managers professionals. We strive to build a quality value-add portfolio in attractive locations for the long term, whilst maintaining a defensive framework. We are buying quality residential assets, we carefully implement various value-add strategies, we deploy premium asset management operations. This results in strong and consistent returns and provides us and our investors with an inflation resistant investment case.

Amalthea prides itself with its long term investors and business partners. We are guided by strict investment principles, including among others, desirable locations, strong population growth, liquid markets, good finance environment, and solid value-add potential.

Рис. 2.7. – Інформаційний блок

Our Projects



Modern House #1


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

📍 Spanndauer Damm 54, 14059 Berlin

Modern House #1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

📍 Spanndauer Damm 54, 14059 Berlin



All projects

Рис. 2.8. – Секція з нашими проектами

Наступний блок(рис.2.7) являє собою такий собі ввідний розділ про сам веб-додаток, а також про те хто ним займається і для чого він потрібен.

Секція що зображена на рис.2.8 це перелік випадкових проектів, з якими ви можете ознайомитися більш детально. Знизу під проектами розташована кнопка за допомогою якої ми переходимо до сторінки з усіма проектами.

Management Team



David Amsalem, Adv.

David is a seasoned entrepreneur with over 20 years of professional real estate experience, focused on real estate investments, projects development and asset management.

David is a strategic advisor for AWZ Ventures Inc. (Canada). Awz Ventures is a Canadian-based global VC firm focused on innovative Israeli, multi-use, AI-powered, deep-tech, cyber security, HLS and intelligence technologies.

David was a senior VP Business Development in one of the leading international real estate development companies active in Europe, and he holds vast experience in deal sourcing, asset acquisition (and disposal), implementation of value-add strategies and creation of joint ventures in Germany, CEE and CIS markets.

Prior to that David worked several years as a commercial lawyer in a leading law firm in Tel Aviv, after he served several years as a commanding officer in the Intelligence Corps of the Israeli army (Captain, Res.).

David holds LL.B degree in Law studies from the Hebrew University in Jerusalem (graduated Magna Cum Laude) and he is a certified lawyer since 2001

[Call Mobile](#) [Email](#)



Hugues Armand-Delille

Hugues has 15 Years of experience, spanning from professional Real Estate to Finance, E-commerce to Crypto, PR & Marketing.

Hugues started his career on Wall Street, working for a leading European Hedge Fund in New York. Following that he co-founded and managed an arts and event community in Paris, still existing today. He later was for several years the managing director of a high growth health e-commerce company in Berlin.

Since 2015, Hugues is completely focused on the development of professional Real Estate Investments business and Asset Management activities, as well as significant Cryptocurrency investments.

Hugues holds a Master Degree in Business Administration from EDC university in Paris (with a major in Finance and honors in Law). He is a Certified NYC Securities Broker (Series 7).

[Call Mobile](#) [Email](#)

Рис. 2.9. – Секція з інформацією про команду менеджерів

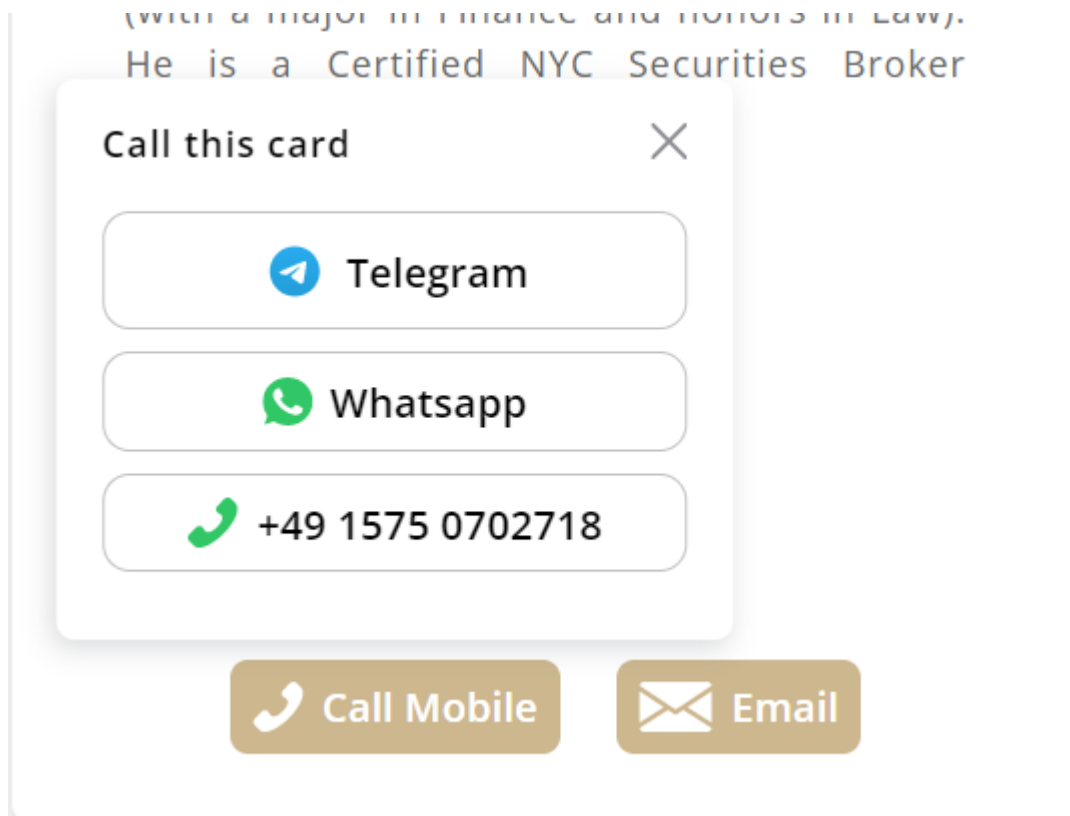


Рис. 2.10. – Способи зв'язку з менеджерами

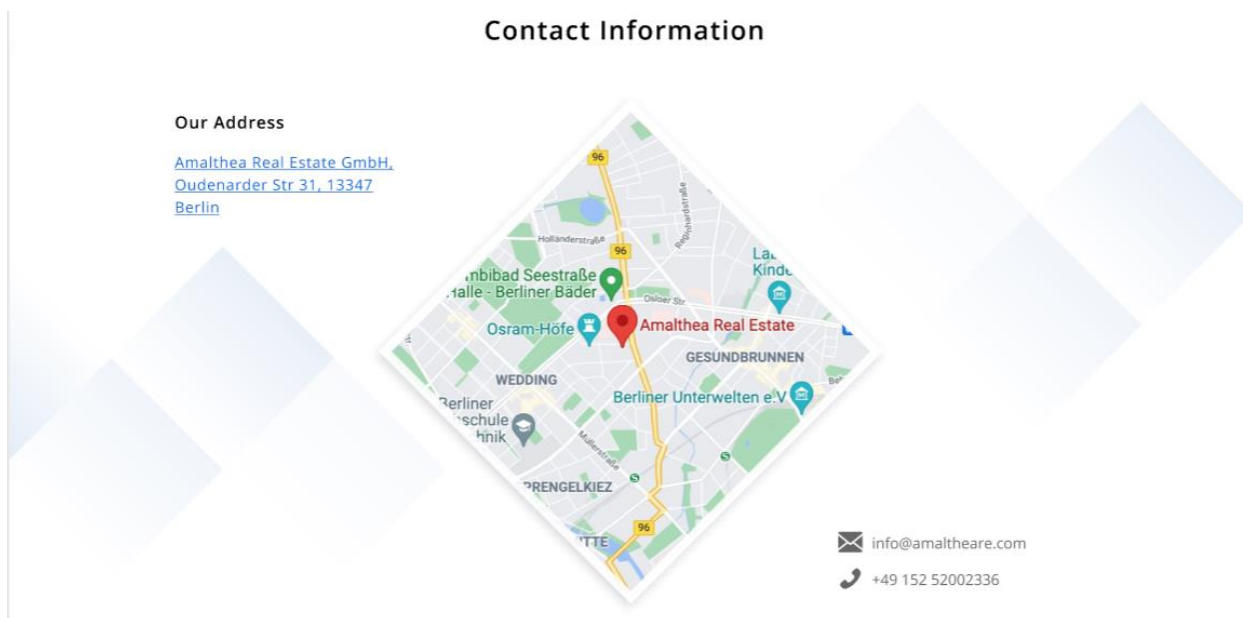
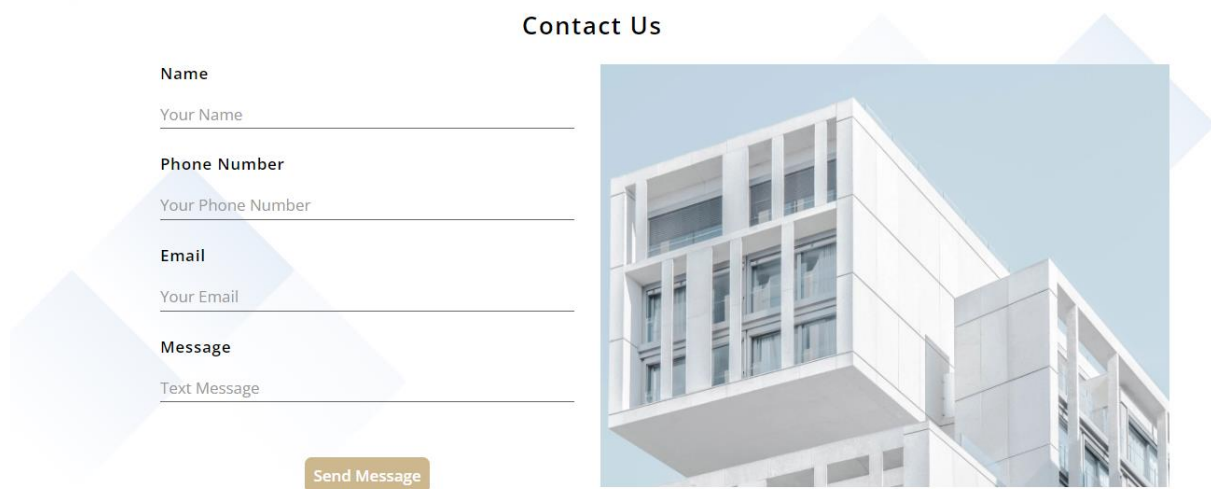


Рис. 2.11. – Способи зв'язку з менеджерами

Далі ми можемо бачити блок (рис.2.9.) у якому нам представлена команда менеджерів які допомагають користувачеві з нерухомістю. Знизу ми

можемо побачити дві кнопки. Перша допомагає користувачеві напряму зателефонувати менеджеру, а друга написати йому листа на електронну пошту. Після того як ми натиснемо на кнопку “Call Mobile” то відкривається рор-up вікно (рис.2.10.) яке надає нам можливість обрати яким саме чином користувач хоче зателефонувати (так як додаток орієнтований на німецького користувача номер телефону менеджера починається на +49). До цього блоку можна швидко перейти через навігаційне меню за посиланням “Teams”.

Також для збільшення можливостей зв’язатися з менеджером, було розміщено додатковий блок, де додатково надається місцезнаходження офісу та контактні дані (рис.2.11).



The image shows a 'Contact Us' form on the left and a photograph of a modern office building on the right. The form has the following fields:

- Name**: Your Name
- Phone Number**: Your Phone Number
- Email**: Your Email
- Message**: Text Message

Below the form is a 'Send Message' button. The office building is a multi-story white structure with large windows and a balcony.

Рис. 2.12. – Форма для зв’язку з адміном

Contact Us

Name
Your Name
field is required

Phone Number
Your Phone Number
field is required

Email
Your Email
field is required

Message
Text Message
field is required

Send Message

Рис. 2.13. – Валідація форми для зв'язку з адміном

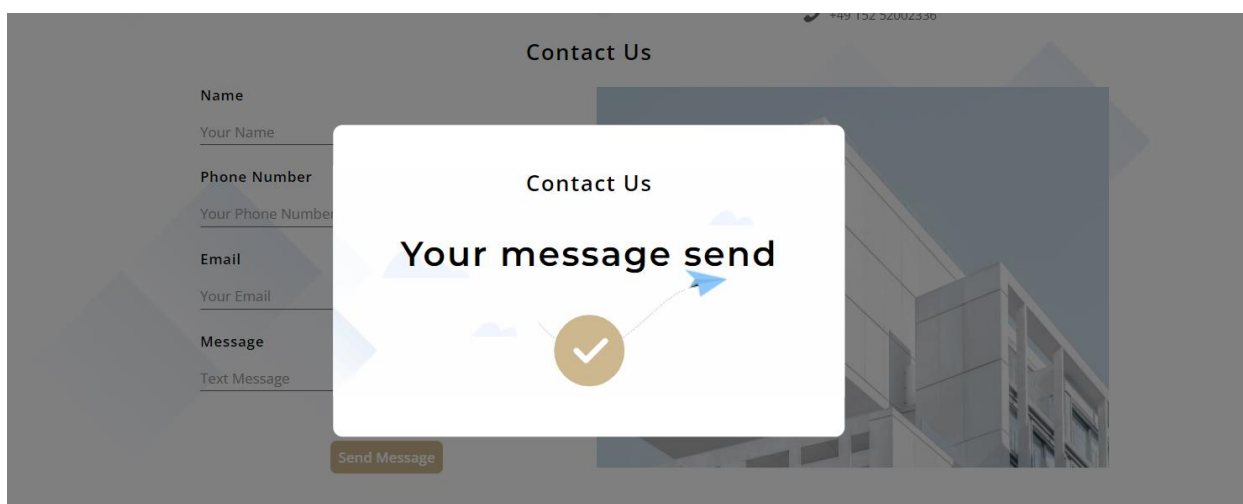


Рис. 2.14. – Ваші дані були правильно заповнені і ваш лист надіслано

Один з пунктів навігаційного меню веде нас до секції під назвою “Contact Us”(рис.2.12) в якій користувач за допомогою заповнення необхідних форм може звернутися напряму до адміністратора веб-додатку з виникнувими питаннями. Кожна форма провалідована, це добре видно на рис.2.13. Валідація запобігає відправленню повідомлення без заповнення всіх необхідних текстових полів. Після успішного заповнення всіх текстових полів, з’явиться рор-уп вікно яке повідомить користувачеві що його повідомлення надіслане.

Imprint

Amalthea Real Estate GmbH

c/o Steuererkontor Fischer, Knesebeckstraße 32, 10623 Berlin, Germany

Register court: Amtsgericht Charlottenburg Register number: HRB 191499 B

VAT number according to § 27a UStG: DE314892487


Authorized representative:

Mr. David Amsalem & Mr. Hugues Armand-Delille

✉ info@amaltheare.com

Рис. 2.15. – Контактні дані


Our Buildings



Modern House #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...


📍 Spanndauer Damm 54, 14059 Berlin

Our Apartments




Modern House #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

📍 Spanndauer Damm 54, 14059 Berlin



Modern House #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

📍 Spanndauer Damm 54, 14059 Berlin



Modern House #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

📍 Spanndauer Damm 54, 14059 Berlin

Рис. 2.16. – Список доступних будівель

Після того як користувач перейшов в навігаційному меню за посиланням “Projects” користувач переходить до сторінки “Our Projects” (рис.2.16.), де всю нерухомість поділено на дві групи. Перша група, це безпосередньо сама нерухомість, а у другій групі знаходяться проекти пов'язані з квартирами.

Біля кожного проекту написано його стан (продається, здається в оренду, у розробці). Цю сторінку також адаптовано під користувачів смартфонів(рис.2.17.) на прикладі iPhone 12 Pro. Також є розподілення на сторінки, на одній сторінці розміщується до чотирьох проектів.

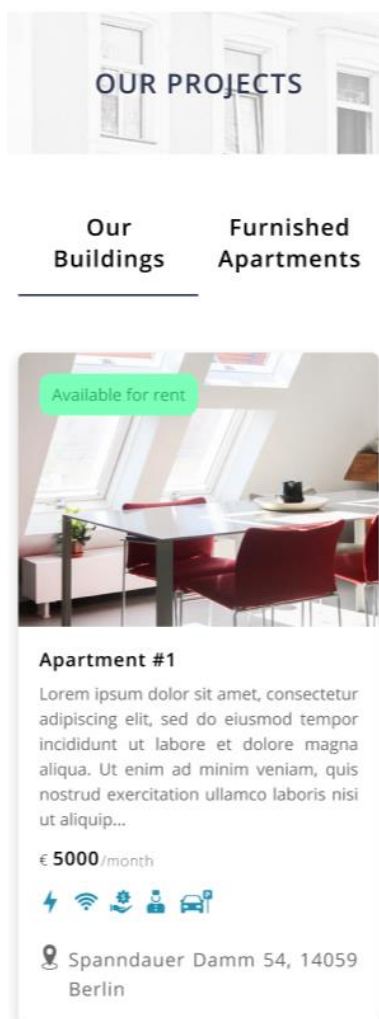


Рис. 2.17. – Список доступних будівель на iPhone 12 Pro

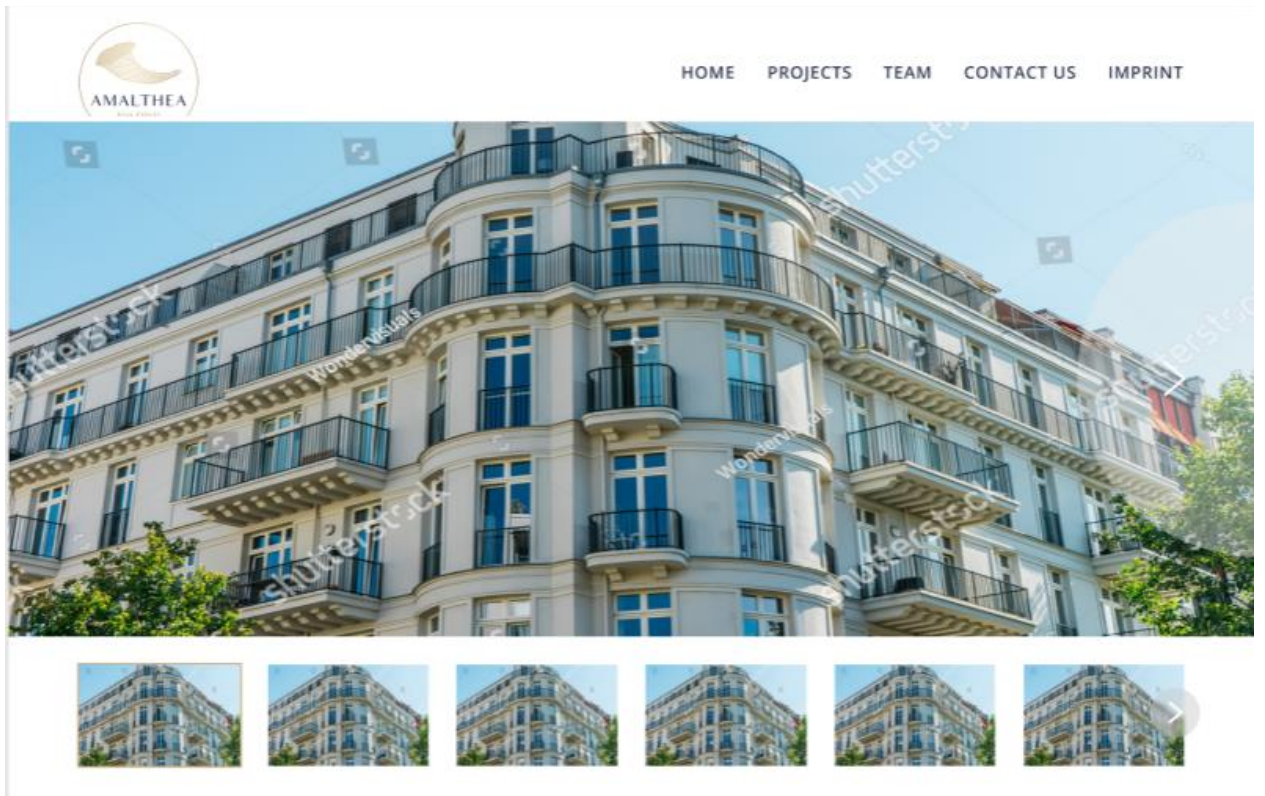


Рис. 2.18. – Слайдер з зображеннями будівлі

Condo In Redwood Village

5 🏠 4,796 📏 Yes 📅 44 🚗 Type 📋 17 🏠 10 📅

Built : 2020 Renovated : 2021

Located in a secluded garden, a stone's throw from two of the world's most pristine white sand beaches is Paradise Cove, where hospitality, and the easy and elegant lifestyle of Anguilla await. Guests enter through the portico and are greeted in the main lobby by genuine, friendly, professional staff. Sitting rooms offer comfort and a focal point to discuss island activities, dinner reservations, individual interests and itinerary requests.

Located in a secluded garden, a stone's throw from two of the world's most pristine white sand beaches is Paradise Cove, where hospitality, and the easy and elegant lifestyle of Anguilla await. Guests enter through the portico and are greeted in the main lobby by genuine, friendly, professional staff. Sitting rooms offer comfort and a focal point to discuss island activities, dinner reservations, individual interests and itinerary requests. From the back side of the lobby, an open-air terrace leads into a garden oasis, the center of Paradise Cove, where an oversized pool with two open-air Jacuzzis are surrounded by palm trees and fragrant gardens.

[🔗](#)

Рис. 2.19. – Дані про будівлю а також кнопка “Поділитися”



Рис. 2.20. – Більш конкретні дані при наведенні на потрібну іконку

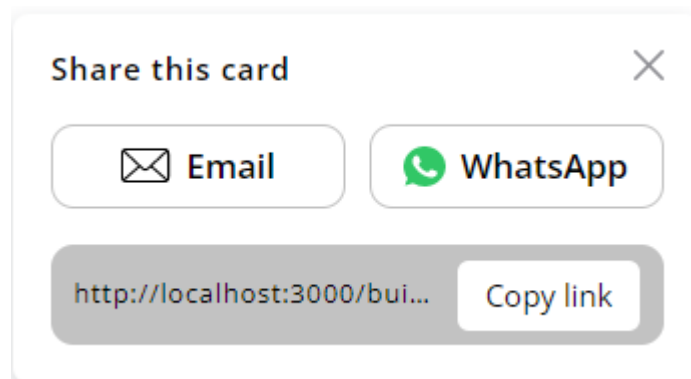


Рис. 2.21. – Способи якими можна поділитися даними про будівлю

Building plan



Рис. 2.22. – План будівлі а також завантаження плану в форматі pdf

Address

Bettinastraße 73, 63067 Offenbach am Main,
Germany

Maria Egides

✉ info@amalthere.com

☎ +49 152 52002336

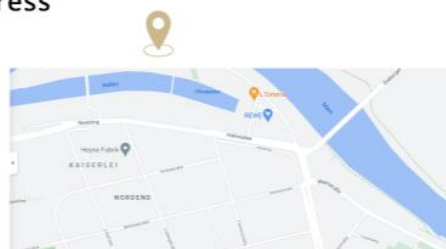



Рис. 2.23. – Місце знаходження будівлі та данні ріелтора

Other Apartments in This Building

Available for rent



Apartment #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 5000 /month

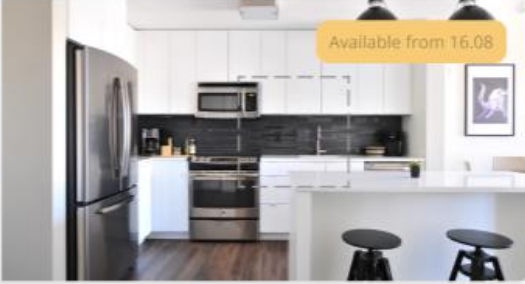
📍 Spanndauer Damm 54, 14059 Berlin

Apartment #2
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 4000 /month

+HK +BK ⚡ 📶 🗑️ 🚿 🚗

📍 Spanndauer Damm 54, 14059 Berlin



Available from 16.08

[See More](#)

Рис. 2.24. – Доступні квартири в даній будівлі

Get More Info About the Building

Name
Your Name

Phone Number
Your Phone Number

Emale
Your Email

Message
Text Message

[Send Message](#)

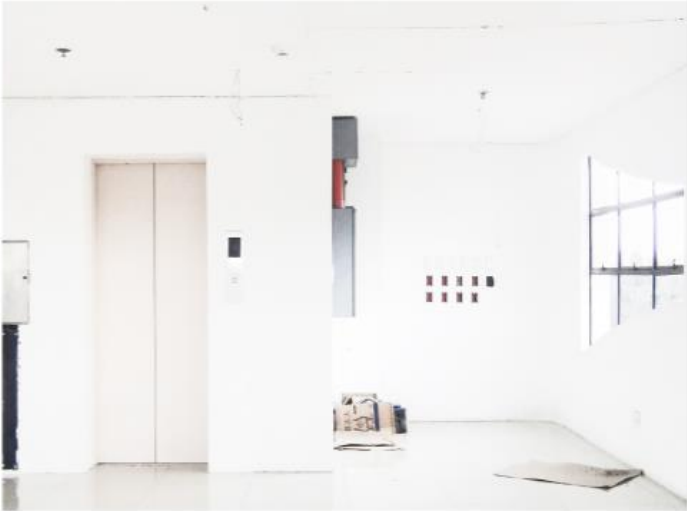



Рис. 2.25. – Форма для зворотного зв'язку

Our Buildings

Our Apartments




Available for rent

Apartment #1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 5000 /month

📍 Spanndauer Damm 54, 14059 Berlin ...



Available from 16.08


Ping An International Financial Center

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 4000 /month

+HK +BK ⚡ 📶 🗺️ 👤 🚗

📍 Spanndauer Damm 54, 14059 Berlin




Sold

Apartment #3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 5500 /month

📍 Spanndauer Damm 54, 14059 Berlin



For sale

Apartment #4

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 6000 /month

+HK +BK ⚡ 📶 🗺️ 👤 🚗

📍 Spanndauer Damm 54, 14059 Berlin

Рис. 2.26. – Каталог доступных квартир

Our Buildings

Our Apartments

Available for rent

Apartment #1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 5000 /month

📍 Spanndauer Damm 54, 14059

Wi-Fi

Ping An International Financial Center

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 4000 /month

+HK +BK ⚡ 📶 🛠️ 🚗

📍 Spanndauer Damm 54, 14059 Berlin

Available from 16.08

Рис. 2.27. – При наведенні на відповідну іконку надає додаткову інформацію

AMALTHEA

HOME PROJECTS TEAM CONTACT US IMPRINT

< >

Thumbnail images: dining table, living room, bedroom, kitchen, office, living room.

Рис. 2.28. – Слайдер для більш детального огляду квартири

Apartment Modern

€ 5,000 /month

4 🛏 3 🚿 11 🧑 107 📏 5 📺 Yes 📺 Yes 🚿 Yes 🧑 Yes 🚿
Yes 📺 Yes 🚿

Located in a secluded garden, a stone's throw from two of the world's most pristine white sand beaches is Paradise Cove, where hospitality, and the easy and elegant lifestyle of Anguilla await. Guests enter through the portico and are greeted in the main lobby by genuine, friendly, professional staff. Sitting rooms offer comfort and a focal point to discuss island activities, dinner reservations, individual interests and itinerary requests.

Located in a secluded garden, a stone's throw from two of the world's most pristine white sand beaches is Paradise Cove, where hospitality, and the easy and elegant lifestyle of Anguilla await. Guests enter through the portico and are greeted in the main lobby by genuine, friendly, professional staff. Sitting rooms offer comfort and a focal point to discuss island activities, dinner reservations, individual interests and itinerary requests. From the back side of the lobby, an open-air terrace leads into a garden oasis, the center of Paradise Cove, where an oversized pool with two open-air Jacuzzis are surrounded by palm trees and fragrant gardens.

Includes: ⚡ 📶 🚿 🧑 🚿



Рис. 2.29. – Детальна інформація про квартири а також її вартість(орента)

Apartment plan



📄 Apartment Plan

📄 Download Expose

Рис. 2.30. – План квартири який можна отримати в форматі pdf

Address

Bettinastraße 73, 63067 Offenbach am Main,
Germany

✉ info@amalthere.com

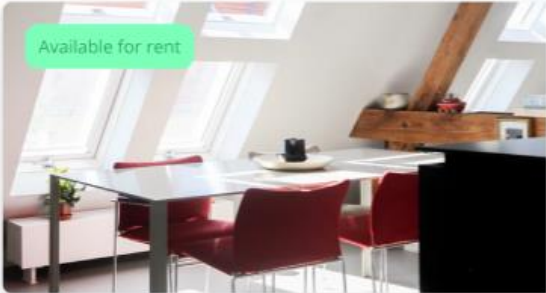
☎ +49 152 52002336



Рис. 2.31. – Місце розташування будівлі а також контакти ріелтора

Other Apartments in This Building

Available for rent



Apartment #1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 5000 /month


📍 Spanndauer Damm 54, 14059 Berlin

Apartment #2
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

€ 4000 /month

+HK +BK ⚡ 📶 🛠️ 🚗

📍 Spanndauer Damm 54, 14059 Berlin



Available from 16.08

See More

Рис. 2.32. – Перелік інших квартир у цій будівлі

Get More Info About the Apartments

Name
Your Name _____

Phone Number
Your Phone Number _____

Emale
Your Email _____

Message
Text Message _____




Рис. 2.33. – Форма для зворотного зв'язку

Після того як користувач переходить на сторінку проекту перше що він бачить це слайдер (рис.2.18., рис.2.28.) з зображенням обраного об'єкту (інтер'єр, екстер'єр). Далі йде опис самого об'єкту, особливості розміщення, переваги, а також в залежності від того в якому стіні зараз знаходиться нерухомість пишеться ціна (рис.2.19., рис.2.29.). Також в цьому блоці присутня кнопка “Поділитися” (рис.2.21., рис.2.29) відкривається рор-уп вікно яке дає змогу зручно ділитися посиланням з іншими людьми зручним для користувача способом.

Наступне, що бачить користувач, це план самої будівлі або квартири з двома кнопками котрі допомагають завантажити користувачеві план будівлі, або сторінку з нерухомість у pdf форматі (рис.2.22., рис.2.30.).

Спускаючись нижче (рис.2.23., рис.2.31), ми бачимо адресу і контактні дані, за допомогою яких ми можемо дізнатися де знаходиться ця квартира або будівля.

Далі йде блок рекомендацій. Якщо ви зайшли на проект з категорії будівлі (рис.2.24.) то вам будуть рекомендуватися для перегляду квартири що є у цій будівлі, а якщо ви зайдете на проект з категорії квартир (рис.2.32), то вам запропоновано інші варіанти квартир у цій будівлі.



Page is not found.

Рис. 2.34. – Сторінка не знайдена(404)

У випадку коли користувач переходить за неправильним посиланням або на неіснуючу сторінку то він потрапляє на 404 сторінку (рис.2.34).

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1600;
2. коефіцієнт складності програми – 1,3;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста – 118 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1;
7. вартість машино-години ЕОМ – 20 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_{\partial}, \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

t_{oml} – витрати праці на налагодження програми на ЕОМ;

t_{δ} – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 1600 \cdot 1,3 \cdot (1 + 0,05) = 2184$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85) \cdot K}, \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

K – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності;

$$t_u = (2184 \cdot 1,2) / (80 \cdot 1) = 32,76 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25) \cdot K}, \quad (3.4)$$

$$t_a = 2184 / (22 \cdot 1) = 99,27 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{q}{(20 \dots 25) \cdot K}, \quad (3.5)$$

$$t_n = 2184 / (20 \cdot 1) = 109,2 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{q}{(4 \dots 5) \cdot K}, \quad (3.6)$$

$$t_{отл} = 2184 / (5 \cdot 1) = 436,8 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{отл}^k = 1,2 \cdot t_{отл}, \quad (3.7)$$

$$t_{отл}^k = 1,2 \cdot 436,8 = 524,16 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначається за формулою:

$$t_d = t_{др} + t_{до}, \text{ людино-годин,}$$

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису:

$$t_{др} = \frac{Q}{(15...20)*k}, \text{ люДИНО-ГОДИН.}$$

$t_{до}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 * t_{д}, \text{ люДИНО-ГОДИН.}$$

Підставляючи відповідні значення, отримуємо:

$$t_{др} = \frac{2184}{15*1} = 145,6, \text{ люДИНО-ГОДИН.}$$

$$t_{до} = 0,75 * 145,6 = 109,2, \text{ люДИНО-ГОДИН.}$$

$$t_{д} = 145,6 + 109,2 = 254,8, \text{ люДИНО-ГОДИН.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 32,76 + 99,27 + 109,2 + 524,16 + 254,8 = 1070,19 \text{ люДИНО-ГОДИН.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ включають витрати на заробітну плату виконавця програми З/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{по} = Z_{зп} + Z_{мв} \text{ грн.}, \quad (3.11)$$

де $Z_{зп}$ – заробітна плата виконавців, яка визначається за формулою:

$$Z_{зп} = t \cdot C_{пр} \text{ грн.}, \quad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$C_{ПР}$ – середня годинна заробітна плата програміста, грн/година

$$З_{ПР} = 1070,19 * 118 = 126282,42 \text{ грн.}$$

$З_{МВ}$ – Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{МВ} = t_{омл} \cdot C_M \text{ грн.}, \quad (3.13)$$

де $t_{омл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{Мч}$ – вартість машино-години ЕОМ, грн/год.

$$З_{мв} = 436,8 * 20 = 8736 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{по} = 126282,42 + 8736 = 135018,42 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.}, \quad (3.14)$$

де B_k - число виконавців;

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

$$T = \frac{1070,19}{1 \cdot 176} \approx 6,1 \text{ міс.}$$

Висновок: програмне забезпечення розроблено для надання користувачу послуг купівлі та оренди нерухомості. Вартість даного програмного забезпечення становить близько 135018,42 грн. Очікуваний час розробки становить 1070,19 годин, тобто 6,1 місяці при 40 годинному робочому тижні. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено практичний та надійний веб-додаток, з використанням сучасних підходів та інструментів розробки.

Головними перевагами використання сучасних підходів до створення програми є зручна подальша масштабованість проекту, за рахунок компонентного підходу розробки веб-додатку.

Пряме призначення даного веб-додатку є надання інформації і контактних даних користувачеві необхідних для придбання або оренди нерухомості, а також необхідні документи для ознайомлення з нерухомістю. Для зручності і практичності веб-додаток дає змогу зберігати документи або сторінку з нерухомістю у форматі pdf.

Веб-застосунок дозволяє користувачу зручно отримувати необхідну йому інформацію, а також дозволяє зручним для користувача методами зв'язуватися з менеджерами які відповідають за ту чи іншу нерухомість, зберігати сторінку необхідної нерухомості, а також зберігати план будівлі.

В результаті отриманий веб-додаток має практичну цінність, тому, що вирішує реальні проблеми такі як заощадження часу пошуку необхідної нерухомості, а також отримання повноти інформації нерухомості.

Під час виконання даної кваліфікаційної роботи були виконані пройдені етапи створення програмного продукту:

- аналіз предметної області задачі, що розв'язується;
- обрання раціональної архітектури та технології створення додатку;
- написання програмного коду веб-додатку;
- розробка рекомендацій щодо використання застосунку;

Під час виконання кваліфікаційної роботи також було визначено трудомісткість розробленого програмного продукту (1070,19 людино-годин), проведений підрахунок вартості роботи по створенню програми (135018,42 грн) та розраховано час на його створення (6.1 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація React. <https://uk.reactjs.org>.
2. Офіційна документація Next.js. <https://nextjs.org/>.
3. Офіційна документація по роботі з формами в React. <https://react-hook-form.com>.
4. Інформаційне онлайн-джерело по роботі з HTML-розміткою <https://www.w3schools.com/html/default.asp>.
5. Онлайн-підручник з вивчення JavaScript. <https://javascript.info>. 1300 с.
6. Довідка по роботі з веб-технологіями. <https://my-js.org/>.
7. Інформаційне онлайн-джерело по роботі з CSS-стилями. <https://developer.mozilla.org/en-US/docs/Web/CSS>.
8. Офіційна документація SASS. <https://sass-lang.com/guide>.
9. Офіційна документація з CSS-modules. <https://github.com/css-modules/css-modules>.
10. Офіційна документація з Git. <https://git-scm.com>.
11. Офіційна документація по Webpack 5. <https://webpack.js.org/concepts/>.
12. Офіційна документація npm. <https://www.npmjs.com>.
13. Офіційна документація з node.js. <https://nodejs.org/en/>.
14. Офіційна документація Axios. <https://axios-http.com/docs/intro>.
15. Офіційна документація по TypeScript. <https://www.typescriptlang.org/>.
16. Офіційна документація babel. <https://babeljs.io>.
17. Офіційний посібник з використання React Google Maps. <https://tomchentw.github.io/react-google-maps/>.
18. Офіційна документація yup. <https://github.com/jquense/yup>.
19. Онлайн-підручник з JavaScript. <https://github.com/getify/You-Dont-Know-JS>.
20. Нік Морган. JavaScript для маленьких, 2021. 368 с.

КОД ПРОГРАМИ

Лістинг context/ModalContext.ts:

```
import { createContext, ReactNode, Dispatch, SetStateAction } from 'react';

const d = (node: ReactNode):void => {};

export interface IModalContext{
  isOpen: boolean;
  open: () => void;
  close: () => void;
  setContent: Dispatch<SetStateAction<ReactNode>>;
  content: ReactNode | null | undefined;
}

export const ModalContext = createContext<IModalContext>({
  isOpen: false,
  open: () => ('open'),
  close: () => {},
  setContent: d,
  content: null,
});
```

Лістинг hooks/modal.hooks.ts:

```
import { useState, useCallback, ReactNode } from 'react';
import { IModalContext } from '../context/ModalContext';

export const useModal = ():IModalContext => {

  const [isOpen, setIsOpen] = useState<boolean>(false);
  const [content, setContent] = useState<ReactNode | null>(null);

  const open = useCallback(() => {
    setIsOpen(true);
  }, []);

  const close = useCallback(() => {
    setIsOpen(false);
  }, []);

  return { isOpen, open, close, setContent, content };
};
```

Лістинг layout/MainLayout/ MainLayout.component.tsx:

```
import { useRouter } from 'next/router'
import cn from 'classnames';
```



```

import {
  Footer,
  Header
} from '../modules';

import { MainLayoutProps } from './MainLayout.types'

import styles from './MainLayout.module.scss';

const MainLayout = ({ children }: MainLayoutProps): JSX.Element => {
  const router = useRouter();
  return (
    <div className={cn(styles.wrapper, {
      [styles.wrapper__second]: router.pathname !== "/",
    })}>
      <Header />
      <div className={styles.content}>
        {children}
      </div>
      <Footer className={styles.footer} />
    </div>
  )
}

export default MainLayout;

```

Лістинг layout/MainLayout/ MainLayout.module.scss:

```

.wrapper {
  display: flex;
  flex-direction: column;
  min-height: 100%;
  &.wrapper__second {
    padding-top: 136px;
  }
}

.content {
  flex: 1 0 auto;
}

.footer {
  flex: 0 0 auto;
}

@media(max-width: 1023px) {
  .wrapper.wrapper__second {
    padding-top: 100px;
  }
}

```

ЛІСТИНГ modules/Footer/Footer.component.tsx:

```
import cn from 'classnames';
import { Logotype, Navigation, Socials, Container } from '../components';

import { FooterProps } from './Footer.types';

import styles from './Footer.module.scss';

const Footer = ({ className, ...props }: FooterProps): JSX.Element => (
  <footer className={cn(className, styles.footer)} {...props}>
    <Container variant="fluid">
      <div className={styles.footer_inner}>
        <Logotype className={styles.footer_logo} variant="dark" />
        <Navigation
          navigation={[
            { label: 'HOME', href: '/' },
            { label: 'IMPRINT', href: '/#imprint' },
            { label: 'Privacy Policy', href: '/policy' }
          ]}
          className={styles.footer_nav}
        />
        <Socials />
      </div>
    </Container>
  </footer>
);
export default Footer;
```

ЛІСТИНГ modules/Header/Header.component.tsx:

```
import { useState, useEffect } from 'react';

import cn from 'classnames';

import { HeaderProps } from './Header.types';

import { Navigation, Logotype, Container } from '../components';

import styles from './Header.module.scss';

const Header = ({ className, ...props }: HeaderProps): JSX.Element => {
  const [menuActive, setMenuActive] = useState<boolean>(false);
  const [fixed, setFixed] = useState<boolean>(false);

  useEffect(() => {
    const handleScrollHeader = () => {
      setFixed(window.pageYOffset > 100);
    };

    window.addEventListener('scroll', handleScrollHeader);
    return () => {
```

```

    window.removeEventListener('scroll', handleScrollHeader);
  };
});

```

```

useEffect(() => {
  menuActive
  ? (window.document.body.style.overflow = 'hidden')
  : (window.document.body.style.overflow = 'auto');
}, [menuActive]);

```

```

return (
  <header
    className={cn(className, styles.header, {
      [styles.active]: menuActive,
      [styles.fixed]: fixed
    })}
    {...props}
  >
    <Container variant="fluid">
      <div className={styles.header_inner}>
        <Logotype
          onClick={() => setMenuActive(false)}
          className={styles.header_logo}
          variant="gold"
        />
        <div className={styles.wrapper}>
          <Navigation
            navigation={[
              { label: 'HOME', href: '/' },
              { label: 'PROJECTS', href: '/buildings/1' },
              { label: 'TEAM', href: '/#team' },
              { label: 'CONTACT US', href: '/#contact-us' },
              { label: 'IMPRINT', href: '/#imprint' }
            ]}
            className={styles.header_nav}
            handleMenuClick={() => setMenuActive(false)}
          />
        </div>
        <div
          className={styles.burger}
          onClick={() => setMenuActive((prev) => !prev)}
        />
      </div>
    </Container>
  </header>
);
};

```

```

export default Header;

```

Лістинг pageComponents/404/404.tsx:

```
import Image from 'next/image';
import { MainLayout } from '../layout';
import { Htag, Container } from '../components';
import image404 from '../public/404/404.png';
import styles from './404.module.scss';

const Page404 = () => (
  <MainLayout>
    <section>
      <Container>
        <div className={styles.wrapper}>
          <div className={styles.image}>
            <Image quality={100} src={image404} />
          </div>
          <Htag className={styles.title} tag="h1" variant="h1">
            Page is not found.
          </Htag>
        </div>
      </Container>
    </section>
  </MainLayout>
);

export default Page404;
```

Лістинг pageComponents/Apartment/Apartment.component.tsx:

```
import Image from 'next/image';
import {
  Button,
  Container,
  Htag,
  Slider,
  ErrorBoundary
} from '../components';
import ContactUsContainer from '../sections/ContactUs/ContactUsContainer';
// import { PDFExport } from '@progress/kendo-react-pdf';
import { Address, Projects, Information } from '../sections';

import { MainLayout } from '../layout';

import styles from './Apartment.module.scss';

import download from '../public/icon/download.svg';
import pdf from '../public/icon/pdf.svg';
import { ApartmentInformation } from '../types';
import { SyntheticEvent } from 'react';
// import PDFApartment from '../components/PDF/PDFApartment/PDFApartment.component';
```

```

export interface ApartmentInformationProps extends ApartmentInformation {
  fetchURL?: (e: SyntheticEvent) => Promise<void>;
  disabledButton?: boolean;
}

```

```

const Apartment = ({
  media,
  mine_apartments,
  link_apartment_plan,
  address,
  location,
  other_apartments,
  relation_contact,
  fetchURL,
  disabledButton,
  ...props
}: ApartmentInformationProps) => {
  // const pdfExportComponent = useRef<any>(null);
  // const handlePDFExport = () => {
  //   pdfExportComponent.current.save();
  // };
  console.log(other_apartments);
  console.log(media);

```

```

return (
  <MainLayout>
    <ErrorBoundary>
      <Slider
        images={media
          ?.filter(({ type }) => type === 'gallery')
          .sort((a, b) => a.position - b.position)}
      />
    </ErrorBoundary>

    <Information
      {...props}
      variant="apartment"
      className={styles.apartment}
    />
    <ErrorBoundary>
      <section className={styles.plan}>
        <Container>
          <div className={styles.plan_inner}>
            <Htag className={styles.plan_title} variant="h2" tag="h2">
              Floor plan
            </Htag>
            <div className={styles.plan_image}>
              {media && media.length ? (
                <img
                  alt="image"
                  src={

```

```

        media?.filter(({ type }) => type === 'building')[0].url ||
        '/apartment/plan.jpg'
    }
  />
): (
  <img alt="image" src={'/apartment/plan.jpg'} />
)
</div>
<div className={styles.plan_buttons}>
  <Button
    href={link_apartment_plan}
    download
    Icon={download}
    target="_blank"
    variant="contained"
    tag="link"
  >
    Floor Plan
  </Button>
  <Button
    disabled={disabledButton}
    Icon={pdf}
    onClick={fetchURL}
    variant="contained"
    tag="button"
  >
    Download Expose
  </Button>
</div>
</div>
</Container>
</section>
</ErrorBoundary>

<ErrorBoundary>
  <Address
    {...location}
    relation_contact={relation_contact}
    address={address}
    className={styles.address}
  />
</ErrorBoundary>

<ErrorBoundary>
  <Projects
    className={styles.projects}
    textButton="See More"
    projects={mine_apartments || other_apartments}
    variant="apartment"
    title={
      other_apartments && other_apartments?.length > 0
    }
  />
</ErrorBoundary>

```

```

        ? 'Other Apartments'
        : 'Other Apartments in This Building'
    }
    href="/apartments/1"
  />
</ErrorBoundary>

<ContactUsContainer
  className={styles.contactUs}
  title="Get More Info About the Apartments"
  image="/contact/apartments.jpg"
  />
</MainLayout>
);
};

export default Apartment;

```

ЛІСТИНГ pageComponents/Building/Building.component.tsx:

```

import {
  Button,
  Container,
  Htag,
  Slider,
  ErrorBoundary
} from '../components';

import ContactUsContainer from '../sections/ContactUs/ContactUsContainer';
// import { PDFExport } from '@progress/kendo-react-pdf';
import { Address, Projects, Information } from '../sections';

import { MainLayout } from '../layout';

import styles from './Building.module.scss';

import download from '../public/icon/download.svg';
import pdf from '../public/icon/pdf.svg';

import { BuildingInformation } from '../types';
import { SyntheticEvent } from 'react';
// import PDFBuilding from '../components/PDF/PDFBuilding/PDFBuilding.component';

export interface BuildingInformationProps extends BuildingInformation {
  fetchURL?: (e: SyntheticEvent) => Promise<void>;
  disabledButton?: boolean;
}

const Building = ({
  address,

```

```

other_apartments,
has_apartments,
link_apartment_plan,
media,
location,
mine_apartments,
relation_contact,
fetchURL,
disabledButton,
...props
}: BuildingInformationProps) => {
  // const pdfExportComponent = useRef<any>(null);

  // const handlePDFExport = () => {
  //   pdfExportComponent.current.save();
  // };

  return (
    <MainLayout>
      <ErrorBoundary>
        <Slider
          images={
            media &&
            media
              .filter(({ type }) => type === 'gallery')
              .sort((a, b) => a.position - b.position)
            }
        />
      </ErrorBoundary>

      <Information
        {...props}
        title="Condo In Redwood Village"
        variant="building"
        className={styles.apartment}
      />
      <ErrorBoundary>
        <section className={styles.plan}>
          <Container>
            <Htag className={styles.plan_title} variant="h2" tag="h2">
              Building plan
            </Htag>
            <Slider
              variant="plan"
              images={
                media &&
                media
                  .filter(({ type }) => type === 'building')
                  .sort((a, b) => a.position - b.position)
                }
            />
          </Container>
        </section>
      </ErrorBoundary>
    </MainLayout>
  );
}

```



```

<div className={styles.plan_buttons}>
  <Button
    Icon={download}
    href={link_apartment_plan}
    download
    variant="contained"
    tag="link"
  >
    Building Plan
  </Button>
  <Button
    disabled={disabledButton}
    onClick={fetchURL}
    Icon={pdf}
    variant="contained"
    tag="button"
  >
    Download Expose
  </Button>
</div>
</Container>
</section>
</ErrorBoundary>

<ErrorBoundary>
  <Address
    relation_contact={relation_contact}
    {...location}
    address={address}
    className={styles.address}
  />
</ErrorBoundary>

<ErrorBoundary>
  <Projects
    className={styles.projects}
    textButton="See More"
    variant="building"
    projects={mine_apartments || other_apartments}
    title={
      other_apartments && other_apartments?.length > 0
        ? 'Other Apartments'
        : 'Other Apartments in This Building'
    }
    href="/apartments/1"
  />
</ErrorBoundary>

<ContactUsContainer
  className={styles.contactUs}
  title="Other Apartments in This Building"

```

```

        image="/contact/apartments.jpg"
      />

    </MainLayout>
  );
};

export default Building;

```

Лістинг pageComponents/Home/Home.component.tsx:

```

import Image from 'next/image';
import Slider from 'react-slick';
import cn from 'classnames';
import {
  Container,
  Htag,
  Paragraph,
  Button,
  ListProject,
  ManagementList,
  Feedback
} from '../components';

import ContactUsContainer from '../sections/ContactUs/ContactUsContainer';

import Message from '../public/icon/message.svg';
import Phone from '../public/icon/phone.svg';

import styles from './Home.module.scss';

import { MainLayout } from '../layout';
import { HomeProps } from '../pages/index';

import Quote from '../components/Quote/Quote.component';

import map from '../public/home/contact/map.png';
import management1 from '../public/content/management/management1.png';
import management2 from '../public/content/management/management2.png';

const Home = ({ buildings }: HomeProps) => (
  <MainLayout>
    <section className={styles.intro}>
      <Slider
        arrows={false}
        autoplay
        speed={3000}
        autoplaySpeed={7000}
        fade
        infinite

```

```

    className={styles.slider}
  >
  <div>
    <div
      style={{ backgroundImage: `url(/home/intro/intro-bg1.png)` }}
      className={styles.intro_bg}
    />
  </div>
  <div>
    <div
      style={{ backgroundImage: `url(/home/intro/intro-bg2.jpg)` }}
      className={styles.intro_bg}
    />
  </div>
  <div>
    <div
      style={{ backgroundImage: `url(/home/intro/intro-bg3.jpg)` }}
      className={styles.intro_bg}
    />
  </div>
  <div>
    <div
      style={{ backgroundImage: `url(/home/intro/intro-bg4.jpg)` }}
      className={styles.intro_bg}
    />
  </div>
</Slider>

<Container variant="fluid">
  <div className={styles.intro_inner}>
    <Htag className={styles.intro_title} tag="h1" variant="h1">
      <span className={styles.intro_light}>WE ARE</span> AMALTHEA
    </Htag>
    <Paragraph className={styles.intro_text} variant="body_1">
      We're a Real Estate Investment company
    </Paragraph>
    <Button href="#contact-us" tag="link" variant="contained">
      Contact Us
    </Button>
  </div>
</Container>
</section>

<section className={styles.property}>
  <Container>
    <Htag className={styles.property_title} tag="h2" variant="h2">
      Our Business Philosophy & Investment Criteria
    </Htag>
    <Paragraph className={styles.property_text} variant="body_1">
      For more than 15 years we have been active in the field of real estate
      investment, project development, and VC investments. Here, with an

```

experienced and dynamic team, we pursue time tested investment cases of strictly grade B+/A+ properties with significant value-add potential in major cities in Germany and Portugal, as well as synergetic investments into related Prop-Tech companies.

</Paragraph>

<Paragraph className={styles.property_text} variant="body_1">

We are experienced real estate and risk managers professionals. We strive to build a quality value-add portfolio in attractive locations for the long term, whilst maintaining a defensive framework. We are buying quality residential assets, we carefully implement various value-add strategies, we deploy premium asset management operations. This results in strong and consistent returns and provides us and our investors with an inflation resistant investment case.

</Paragraph>

<Paragraph className={styles.property_text} variant="body_1">

Amalthea prides itself with its long term investors and business partners. We are guided by strict investment principles, including among others, desirable locations, strong population growth, liquid markets, good finance environment, and solid value-add potential.

</Paragraph>

</Container>

</section>

<section id="projects" className={styles.property}>

<Htag className={styles.property_title} tag="h2" variant="h2">

Our Projects

</Htag>

<Container>

<ListProject

variant="building"

className={styles.projects_list}

projects={buildings}

/>

<div className={styles.projects_button}>

<Button

className={styles.projects_button}

variant="contained"

tag="link"

href="/buildings/1"

>

All projects

</Button>

</div>

</Container>

</section>

<section id="team" className={styles.managementTeam}>

<Container>

<Htag className={styles.projects_title} tag="h2" variant="h2">

Management Team

</Htag>

```
<ManagementList
managements={[
  {
    email: 'mailto:info@amalthoare.com',
    image: management1,
    phone: 'tel:4915252002336',
    text: `David is a seasoned entrepreneur with over 20 years of professional real estate
experience, focused on real estate investments, projects development and asset management.
```

David is a strategic advisor for AWZ Ventures Inc. (Canada). Awz Ventures is a Canadian-based global VC firm focused on innovative Israeli, multi-use, AI-powered, deep-tech, cyber security, HLS and intelligence technologies.

David was a senior VP Business Development in one of the leading international real estate development companies active in Europe, and he holds vast experience in deal sourcing, asset acquisition (and disposal), implementation of value-add strategies and creation of joint ventures in Germany, CEE and CIS markets.

Prior to that David worked several years as a commercial lawyer in a leading law firm in Tel Aviv, after he served several years as a commanding officer in the Intelligence Corps of the Israeli army (Captain, Res.).

David holds LL.B degree in Law studies from the Hebrew University in Jerusalem (graduated Magna Cum Laude) and he is a certified lawyer since 2001

```
,
title: 'David Amsalem, Adv.',
contact: {
  phone: {
    label: '+49 1525 2002336',
    value: 'tel:4915252002336'
  },
  telegram: {
    label: 'Telegram',
    value: 'https://t.me/DavidBerlin'
  },
  whatsapp: {
    label: 'Whatsapp',
    value: 'https://api.whatsapp.com/send?phone=4915252002336'
  }
}
},
{
  image: management2,
  email: 'mailto:info@amalthoare.com',
  phone: 'tel:4915252002336',
  text: `Hugues has 15 Years of experience, spanning from professional Real Estate to
Finance, E-commerce to Crypto, PR & Marketing.
```

Hugues started his career on Wall Street, working for a leading European Hedge Fund in New York. Following that he co-founded and managed an arts and event community in Paris,

still existing today. He later was for several years the managing director of a high growth health e-commerce company in Berlin.

Since 2015, Hugues is completely focused on the development of professional Real Estate Investments business and Asset Management activities, as well as significant Crypto-currency investments.

Hugues holds a Master Degree in Business Administration from EDC university in Paris (with a major in Finance and honors in Law). He is a Certified NYC Securities Broker (Series 7).

```
\n,\ntitle: 'Hugues Armand-Delille',\ncontact: {\n  phone: {\n    label: '+49 1575 0702718',\n    value: 'tel:4915750702718'\n  },\n  telegram: {\n    label: 'Telegram',\n    value: 'https://t.me/the_hugi'\n  },\n  whatapp: {\n    label: 'Whatsapp',\n    value: 'https://api.whatsapp.com/send?phone=4915750702718'\n  }\n}\n}\n}\n]]\n/>\n</Container>\n</section>
```

```
<section>\n  <Htag className={styles.contactTitle} variant="h2" tag="h1">\n    Contact Information\n  </Htag>\n  <div className={styles.contact}>\n    <Container>\n      <div className={styles.contact_inner}>\n        <div className={styles.contact_address}>\n          <Htag\n            className={styles.contact_addressTitle}\n            variant="title_1"\n            tag="h5"\n          >\n            Our Address\n          </Htag>\n          <a\n            target="_blank"\n            href="https://www.google.com.ua/maps/place/Amalthea+Real+Estate/@52.5532797,13.3628168,17z/data=!4m5!3m4!1s0x47a85368e818ed0d:0x66314ac47e0e35a5!8m2!3d52.5533876!4d13.3649481?hl=en"
```

```

        className={styles.contact_addressLink}
      >
        Amalthea Real Estate GmbH, Oudenarder Str 31, 13347 Berlin
      </a>
    </div>
    <div>
      <a
        className={styles.map}
        target="_blank"
        href="https://www.google.com/maps/place/Oudenarder+Str.+31,+13347+Berlin,+%D
0%93%D0%B5%D1%80%D0%BC%D0%B0%D0%BD%D0%B8%D1%8F/@52.553326,13.36
27604,17z/data=!4m5!3m4!1s0x47a8522c7ef4c82d:0x9e50be18c3bb8044!8m2!3d52.553326!4d
13.3649491"
      >
        <Image
          alt="image"
          quality={100}
          src="/home/contact/map.png"
          height={511}
          width={511}
        />
      </a>
    </div>
    <div className={styles.contact_links}>
      <Feedback Icon={Message} href="mailto:info@amaltheare.com">
        { ` info@amaltheare.com` }
      </Feedback>
      <Feedback Icon={Phone} href="tel:4915252002336">
        { ` +49 152 52002336` }
      </Feedback>
    </div>
  </div>
</Container>
</div>
</section>
<ContactUsContainer
  id="contact-us"
  image="/contact/building.jpg"
  title="Contact Us"
></ContactUsContainer>

<section id="imprint" className={styles.imprint}>
  <Container>
    <Htag className={styles.imprint_title} variant="h2" tag="h1">
      Imprint
    </Htag>
    <div className={styles.imprint_inner}>
      <div className={styles.imprint_start}>
        <Htag
          className={styles.imprint_titleSecond}
          tag="h5"

```

```

    variant="title_1"
  >
    Amalthea Real Estate GmbH
  </Htag>
  <Paragraph className={styles.imprint_text} variant="body_2">
    C/o Steuererkontor Fischer, Knesebeckstraße 32, 10623 Berlin,
    Germany
  </Paragraph>
  <Feedback
    href="mailto:info@amaltheare.com"
    className={styles.imprint_link}
    Icon={Message}
  >
    <Image
      alt="image"
      quality={100}
      src={map}
      height={511}
      width={511}
    />
  </Feedback>
</div>
<div className={styles.contact_links}>
  <Feedback Icon={Message} href="mailto:info@amaltheare.com">
    {` info@amaltheare.com`}
  </Feedback>
</div>
<div className={styles.imprint_end}>
  <Htag
    className={styles.imprint_titleSecond}
    tag="h5"
    variant="title_1"
  >
    Authorized representative:
  </Htag>
  <Paragraph className={styles.imprint_text} variant="body_2">
    Mr. David Amsalem & Mr. Hugues Armand-Delille
  </Paragraph>
  <Paragraph className={styles.imprint_text} variant="body_2">
    Register court: Amtsgericht Charlottenburg Register number: HRB
    191499 B
  </Paragraph>
  <Paragraph className={styles.imprint_text} variant="body_2">
    VAT number according to § 27a UStG: DE314892487
  </Paragraph>
</div>
</div>
</Container>
</section>
<ContactUsContainer
  id="contact-us"

```



```
image="/contact/building.jpg"
title="Contact Us"
></ContactUsContainer>
```

```
<section id="imprint" className={styles.imprint}>
  <Container>
    <Htag className={styles.imprint_title} variant="h2" tag="h1">
      Imprint
    </Htag>
    <div className={styles.imprint_inner}>
      <div className={styles.imprint_start}>
        <Htag
          className={styles.imprint_titleSecond}
          tag="h5"
          variant="title_1"
        >
          Amalthea Real Estate GmbH
        </Htag>
        <Paragraph className={styles.imprint_text} variant="body_2">
          c/o Steuererkontor Fischer, Knesebeckstraße 32, 10623 Berlin,
          Germany
        </Paragraph>
        <Paragraph className={styles.imprint_text} variant="body_2">
          Register court: Amtsgericht Charlottenburg Register number: HRB
          191499 B
        </Paragraph>
        <Paragraph className={styles.imprint_text} variant="body_2">
          VAT number according to § 27a UStG: DE314892487
        </Paragraph>
      </div>
      <div className={styles.imprint_end}>
        <Htag
          className={styles.imprint_titleSecond}
          tag="h5"
          variant="title_1"
        >
          Authorized representative:
        </Htag>
        <Paragraph className={styles.imprint_text} variant="body_2">
          Mr. David Amsalem & Mr. Hugues Armand-Delille
        </Paragraph>

        <Feedback
          href="mailto:info@amaltheare.com"
          className={styles.imprint_link}
          Icon={Message}
        >
          {` info@amaltheare.com `}
        </Feedback>
      </div>
    </div>
  </div>
```

```

    </Container>
  </section>
</MainLayout>
);

export default Home;

```

Лістинг pageComponents/Projects/Projects.component.tsx:

```

import cn from 'classnames';
import Link from 'next/link';

import {
  Banner,
  Container,
  ListProject,
  Pagination,
  ErrorBoundary
} from '../../components';
import { MainLayout } from '../../layout';

import { ProjectsProps } from './Projects.types';

import styles from './Projects.module.scss';

const Projects = ({
  data,
  pathname,
  variant,
  meta: { current_page, last_page },
  onPageChange
}: ProjectsProps) => {
  return (
    <MainLayout>
      <Banner
        className={styles.banner}
        image={'/projects/banner-bg.jpg'}
        title="OUR PROJECTS"
      />
      <Container>
        <div className={styles.panel}>
          <Link href="/buildings/1" passHref>
            <a
              className={cn(styles.link, {
                [styles.active]: pathname.includes('buildings')
              })}
            >
              Our Buildings

```

```

    </a>
  </Link>
  <Link href="/apartments/1">
    <a
      className={cn(styles.link, {
        [styles.active]: pathname.includes('apartments')
      })}
    >
      Our Apartments
    </a>
  </Link>
</div>
<ErrorBoundary>
  <ListProject
    className={styles.projectList}
    variant={variant}
    projects={data}
  />
</ErrorBoundary>
<Pagination
  className={styles.project_pagination}
  onPageChange={onPageChange}
  initialPage={current_page - 1}
  pageCount={last_page}
  marginPagesDisplayed={1}
  pageRangeDisplayed={2}
  disableInitialCallback
/>
</Container>
</MainLayout>
);
};

export default Projects;

```

Всі інші файли з програмним кодом можна буде переглянути на магнітному носії.

ВІДГУК
керівника економічного розділу
на кваліфікаційну роботу бакалавра
на тему:

"Створення веб-орієнтованого додатку з надання послуг купівлі та здачі
в оренду нерухомості на React.js."
студента групи 121-18-2 Демиденка Вадима Андрійовича

Керівник економічного розділу
доцент каф. ПЕП та ПУ, к.е.н

Л. В. Касьяненко

Перелік файлів на диску

ПЕРЕЛІК ДОКУМЕНТІВ НА МАГНІТНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота Демиденко В.А.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота Демиденко В.А.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Демиденко В.А.rar	Архів. Містить коди програми і скомпільовану програму
Презентація	
Демиденко В.А.ppt	Презентація кваліфікаційної роботи