

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота**  
**на здобуття ступеня бакалавра**  
за спеціальністю 121 Програмна інженерія  
на тему:

**РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ СТВОРЕННЯ ТА РЕДАГУВАННЯ  
МЕДІАКОНТЕНТУ**

Виконав студент 4-го курсу  
Максим ГРИГОРЧУК



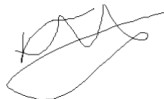
Науковий керівник:

доцент, кандидат фіз.-мат. наук

Лариса КАТЕРИНИЧ \_\_\_\_\_ (підпис)

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

Студент (підпис)



Роботу розглянуто й допущено до захисту  
на засіданні кафедри інтелектуальних  
програмних систем

«\_\_\_» \_\_\_\_\_ 2022 р.,

протокол No \_\_\_\_

Завідувач кафедри

Олександр ПРОВОТАР (підпис)

## РЕФЕРАТ

Обсяг роботи: 23 сторінок, кількість ілюстрацій: 7, використані джерела: 11.

### РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ СТВОРЕННЯ ТА РЕДАГУВАННЯ МЕДІАКОНТЕНТУ.

Мета роботи: створення веб-додатку, який надає можливість користувачам вивчити та познайомитися з новими думками про створення ефективних маркетингових матеріалів, або ж замовити вже готовий варіант рекламного ролику або іншого виду маркетингової продукції.

Методи розробки: розбиття на компоненти великого проекту, використання react hooks та react state management.

Інструменти розробки: Середовище розробки WebStorm, мова програмування JavaScript, бібліотеки ReactJS, React Redux та React Query.

Результати роботи: виконано загальний огляд технологій та підходів, що можуть бути реалізовані з метою вирішення даної задачі. На основі переваг та недоліків розглянутих засобів було сформульована стратегія реалізації та був розроблений веб-ресурс, який задовільняє всі завдання, які були поставлені на початку проекту.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	4
ВСТУП.....	5
РОЗДІЛ 1. ПЕРЕДУМОВИ СТВОРЕННЯ ІНСТРУМЕНТА.....	8
1.1 Актуальність теми.....	8
1.2 Розширений огляд теми.....	8
1.3 Обрання інструментів та технологій розробки.....	9
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА.....	11
2.1 База даних проекту Google Firestore.....	11
2.2 Структура бази даних проекту.....	12
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ.....	17
3.1 Поділ на основні компоненти.....	17
3.2 Компонент Homepage.....	17
3.4 Компоненти Blog та Admin.....	20
ВИСНОВКИ.....	22

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ**

JS — JavaScript.

IDE — Integrated Development Environment, інтегроване середовище розробки.

VCS — Version Control System, система контролю версій.

API — Application Programming Interface, інтерфейс прикладного програмування.

SQL — Structured Query Language, декларативна мова програмування, яка дозволяє взаємодіяти з базами даних.

JSON — JavaScript Object Notation, формат передачі даних.

URL — Uniform Resource Locato, вказівник на ресурс.

## ВСТУП

**Оцінка сучасного стану об'єкта розробки. Актуальність роботи та підстави для її виконання.** В ХХІ столітті, коли кожного року активно розвиваються сотні нових компаній, створюються величезна кількість стартапів та винаходяться унікальні додатки у абсолютно всіх сферах, роль реклами важко недооцінити, оскільки дуже часто той продукт, рекламна стратегія якого була найуспішнішою, часто стає найбільш уживаним та відомим. Тому зараз набувають дуже великої популярності такі професії, як маркетолог та креативний менеджер. Але створення рекламних матеріалів — це непростий процес, який потребує багатьох знань, умінь та можливостей. Саме цьому у мене виникла ідея сприяти розвитку української компанії по створенню відеоконтенту та навчанню інших людей у цьому напрямку. Наразі існують декілька іноземних аналогів, які публікують дуже багато інформації, але всі їхні зусилля спрямовані на те, щоб користувач купив у них той продукт, який вони намагаються йому продати, що не є найзручнішим підходом у вивченню будь-якої сфери, оскільки часто можна наткнутися на те, що простіше декілька разів переплатити якійсь організації, щоб створити рекламний ролик, ніж навчити цьому когось.

**Мета і завдання роботи.** За мету роботи взято потребу створити такий ресурс, який зможе коротко та дуже точно описувати нагальні проблематики створення маркетингових продуктів і точно передавати шляхи виправлення їх недоліків. Також даний продукт повинен надавати інформацію компаніям, які хочуть найняти провідних фахівців для аналізу та створення певних матеріалів, які можуть використовуватися даною компанією не тільки у маркетингових, але й у інших цілях.

Даний ресурс повинен бути максимально простим у своєму застосуванні, щоб будь-яка людина, яка цікавиться даною тематикою мала змогу використовувати запропонований веб-сайт без довгого ознайомлення з

його функціоналом та можливостями. Також даний інструмент повинен бути доступним з будь-яких девайсів при будь-яких параметрах екранів та різних операційних системах.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- обрати найзручніші та найшвидші технології написання даного ресурсу;
- спроектувати базу даних для зберігання, створення та редагування всіх навчальних матеріалів;
- розробити зрозумілу веб-частину для ознайомлення з вже готовими медіаресурсами та блогом з корисними навчальними статтями;
- спроектувати та розробити адмін-сторінку для максимально гнучкого створення цих статей;
- протестувати отриману систему.

**Об'єкт, методи й засоби розробки.** Об'єктом розробки є спрощення процесу створення та отримання нової інформації про ефективні маркетингові відеоматеріали.

Для розробки було обрано наступні засоби та інструменти:

- В якості мови розробки було обрано JavaScript з бібліотекою ReactJs 17.0.2;
- В якості бібліотеки для створення та роботи з глобальним станом ресурсу було використано React Redux 7.2.6;
- В якості бібліотеки для роботи з асинхронними даними було використано React Query 3.39.0;

- В якості IDE (інтегрованого середовища розробки) було обрано WebStorm 2022.1;
- В якості VCS (системи контролю версій) використано Git, для зберігання коду на віддаленому сервері обрано GitHub.

**Можливі сфери застосування.** Створений ресурс може стати у нагоді тим, хто хоче покращити свої знання з створення маркетингових продуктів або для тих стартапів або компаній, які не мають бажання наймати окремого працівника, проте хочуть створити унікальні та ефективні ролики рекламні матеріали для використання у комерційних цілях.

## **РОЗДІЛ 1. ПЕРЕДУМОВИ СТВОРЕННЯ ІНСТРУМЕНТА**

### **1.1 Актуальність теми**

Реклама відіграє дуже важливу роль у розвитку нових компаній. Світ бізнесу є конкурентним, і реклама використовується, щоб представити бізнес, створити бренд і позиціонувати компанію, продукт чи послугу проти конкурентів. Реклама забезпечує стратегічний обмін повідомленнями та підвищує обізнаність на даному ринку. Для доставки реклами на ринок використовується кілька рекламних носіїв.

Підприємства можуть отримати доступ до аудиторії за допомогою реклами через різноманітні засоби масової інформації. Радіо, телебачення, друк та цифрова реклама є великими каналами для рекламодавців. Також існує безліч креативних варіантів, таких як реклама бензонасоса, реклама у ванних кімнатах та інші привабливі медіа. Радіо, телебачення та друкована преса вважаються традиційними ЗМІ, тоді як цифрові охоплюють все онлайн. У цифровому просторі реклама надходить через потокове радіо і телебачення, платний пошук, соціальну рекламу та медійну або банерну рекламу. Усі ці засоби масової інформації дають змогу компанії розміщувати свою рекламу та вимірювати реакцію та збільшення продажів у результаті реклами. Деякі цифрові медіа пропонують надзвичайно детальні дані для вимірювання не лише відповіді, а й поведінки цих респондентів. Ці дані потім використовуються для покращення платформи продажу для збільшення конверсій на основі успішних поведінкових рис.

### **1.2 Розширений огляд теми**

Відеореклама об'єднує дві речі, які привертають увагу користувачів: рух і звук. Обидва вони відіграють важливу роль у передачі ефективного



повідомлення, що призводить до кращої взаємодії. Аналітична компанія MediaMind [1] проаналізувала понад три мільярди показів реклами в усьому світі за шість місяців, вони виявили, що шанс на те, що користувач натисне на відеорекламу в Інтернеті у 27.4 рази більше, ніж у стандартних банерів.

Якщо ж графічні або текстові оголошення покладаються на копію, пунктуацію та візуальні ознаки, щоб передати потрібний тон, то відеореклама використовує рух і звук, щоб привернути увагу відвідувачів. Формат оголошення ефективніший у донесенні вашої розповіді до цільової аудиторії за допомогою спеціального повідомлення, яке їй легко сприйняти.

Відеореклама, яка викликає інтерес у користувачів і закінчується переконливим остаточним заклик до дії, обов'язково матиме успіх у підтримці зацікавленості аудиторії.

Дослідження показують, що середній користувач запам'ятовує 95% повідомлення, коли його переглядають, порівняно з 10% коли його читають. Це означає, що потенційні клієнти з більшою ймовірністю запам'ятають повідомлення компанії за допомогою відеореклами, а не графічної чи текстової реклами. Крім того, позитивне враження користувача від взаємодії з відеорекламою збільшує його намір купити продукт компанії на 97%. [2] Тому дуже важливо правильно створювати маркетингові відеоматеріали для ефективного популяризування певної компанії серед її конкурентів.

### **1.3 Обрання інструментів та технологій розробки**

Для написання веб-сторінки я обрав бібліотеку ReactJs, оскільки вона надає ресурси для розробки багаторазових і легко інтегрованих компонентів інтерфейсу користувача, які скорочують час розробки та забезпечують адаптивний інтерфейс користувача для веб-сайтів, веб-додатків і мобільних додатків. Також ReactJs пропонує велику колекцію компонентів інтерфейсу,

які можна легко використовувати для створення персоналізованих інтерактивних елементів. [3]

В якості бази даних була обрана Google Firestore — NoSQL база даних, яка дуже зручна у використанні, особливо для невеликих проектів.

Для глобального стану проекту було обрано React Redux та оскільки я є прихильником розділення асинхронних даних та звичайний у глобальному стані, то для обробки асинхронних даних будемо використовувати React Query. Комбінація цих двох бібліотек дозволяє дуже зручно взаємодіяти з даними будь-якого виду та типу та мати легкий доступ до них у всіх компонентах проекту, незалежно від того, які зв'язки між цими компонентами та даними. [4]

Логічним вибором IDE, при усіх використаних бібліотеках для JS, став WebStorm від JetBrains, як один з найкращих серед аналогів для даної мови програмування. Використання професійної версії стало можливим завдяки тому, що компанія JetBrains надає безкоштовний доступ усім студентам вищих навчальних закладів. [5]

## РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

### 2.1 База даних проекту Google Firestore

Cloud Firestore - це новітня база даних Firebase для розробки мобільних додатків з унікальної моделлю даних, що забезпечує просту та інтуїтивну взаємодію з користувачем. Cloud Firestore пропонує:

- зберігання даних у JSON форматі;
- взаємодія з Cloud Storage, що гарантує зберігання, оновлення та легкий доступ до даних;
- можливість користування з декількох пристроїв;
- клієнтські SDK без серверів для розгортання та обслуговування;
- оновлення в реальному часі;
- доступний рівень безкоштовного користування, що передбачає оплату в подальшому за послуги, що використовуються користувачем

Описаний вище список функцій стає доступним одразу після реєстрації, як і багато інших додаткових можливостей. Існує платна версія, що передбачає ширший інструментарій, але безкоштовної версії більш ніж достатньо для створення веб і мобільних платформ. Cloud Firestore пропонує стандартний набір типів даних: array, boolean, bytes, date&time, floating-point number, geographical point, integer, map, null, reference, text string. [6]

NoSQL є революційною альтернативою реляційним базам даних, в яких дані розміщені у таблицях, а їхня структура послідовно розробляється перед створенням. Бази даних NoSQL мають масштабований, високопродуктивний та гнучкий характер.

Цей підхід також може мати справу з різноманітними моделями даних, оскільки у NoSQL DB дані зберігаються у вигляді колекцій документів. Прості дані зберігаються в форматі, подібному до JSON. Дані зі складною ієрархічною структурою можна організувати, використовуючи вкладені колекції в документах.

Основні переваги NoSQL database:

- високий рівень гнучкості
- реплікація – копіювання даних під час оновлення на інші сервери
- шаринг – це поділ інформації з різних вузлів мережі, коли кожен вузол відповідає за визначений набір даних і обробляє запити, що стосуються виключно цього набору даних.
- здатність опрацьовувати великі дані
- масштабованість, що не потребує збільшення апаратного забезпечення

Серед недоліків виділяють основну особливість NoSQL – неявну схему даних, що може призводити до хаотичного зберігання даних та неминучого рефакторингу. Цього можна запобігти, якщо схему даних було ретельно спроектовано та покрито необхідними тестами задля перевірки форматів. [7]

## **2.2 Структура бази даних проекту**

В даній базі даних повинен зберігатися текст та посилання, які знаходиться на всіх сторінках веб-ресурсу, щоб у подальшому у адмінів проекту була можливість редагувати даний текст або додавати абсолютно нові його частини. Цей текст повинен зберігатися двома мовами: українською та англійською, з подальшим можливим додаванням інших мов. Також потрібно зберігати всі створені навчальні статті з можливістю їх редагування та створення нових.

Щоб задовільнити дані потреби проекту структура бази даних буде виглядати так:

1 Текст на посилання українською

- словник, де ключі — це назви сторінок, а значення — це інформація на сторінці (текст, посилання, логотипи, і т.д.)

2 Текст та посилання англійською

- структура в більшості повторюється з першим пунктом

3 Інформація про всі статті

- статті навчального типу
- статті ознайомлювального типу
- загальні статті

### 2.1.3 Структура зберігання статті у базі даних

Оскільки стаття може містити в собі абсолютно різні елементи: текст різного форматування, зображення, відео, гіф-файл, цитату, тощо, то буде зберігати одну статтю, як масив з блоків.

Розглянемо, що таке блок. Блок — це поділ статті на горизонтальні смужки, де кожний блок — це окремий індивідуальний компонент, який не залежить від інших, але при виведенні на екран, всі ці блоки виглядають як одне ціле та утворюють статтю, як це показано на Рис 2.1.

Блок 1
Блок 2
Блок 3

Блок 4

Рис 2.1 Схеми розбиття статті на блоки

Кожен блок може бути поділений на довільну кількість клітинок, де клітинка — це найменша умовна одиниця, яка складається з лише одного типу даних. Наприклад, перша клітинка — це лише текст, а друга картинка — це лише зображення. В кожній з цих клітинок можна редагувати її ширину відносно блоку, в якому вона знаходиться, та її контент, тобто змінювати її текст, відео, тощо.

Отже, кінцевий вигляд структури статті буде такий, як показано на Рис 2.2.

Текст 1.1	Зображення 1.2	Текст 1.3
Відео 2.1		
Зображення 3.1	Текст 3.2	
Текст 4.1	Текст 4.2	Зображення 4.3

Рис 2.2. Схеми розбиття статті на блоки та клітини

Даний тип представлення статті дозволяє створити максимально гнучкі умови їх створення з подальшим простим редагуванням та доданням чогось нового. Адже, як тільки адмін захотів додати щось нове, він може додати новий блок, повністю налаштувати його та зберегти внесені зміни. Також через таке представлення дуже просто можна зберігати статтю у базі даних, як масив блоків, а блок, як масив клітинок, де клітинка має свої налаштування, такі як ширина, тип контенту, тощо та сам контент: текст, посилання зображення або відео.

## 2.3 Оперування даними у глобальному стані проекту

При використанні Redux разом з React у проекті, глобальні дані, які ми використовуємо у одному компоненті більше не потрібно буде передавати у всі інші компоненти, де ми їх використовуємо. Це суттєво полегшує взаємодію всіх компонентів, як між собою, так і з даними, які ці компоненти потребують, особливо якщо ці дані у певних випадках співпадають.

Структура React Redux поділена на 3 частини:

- Store — глобальний стан, де можуть зберігатися всі дані, які є спільними для різних компонентів. Дані в store можуть бути додані, видалені та відредаговані за потребою. [8]
- Action — дія, яка вказує на те, що ми прагнемо щось змінити у store та передаємо спеціальний ідентифікатор, що саме буде означати ця дія: будь-яка обробка даних чи їх зміна, додавання нових та змінення цілих структур store.
- Reducer — функція, яка відловлює те, коли стався якийсь action та виконує ту дію, яка співпадає з ідентифікатором в action, який був викликаний. [9]

## 2.4 Використання React Query

Оскільки у даному додатку ми будемо мати справу з багатьма даними з бази даних, то більшість даних, які у нас будуть зберігатися у store будуть асинхронними, тобто дані, які не синхронізуються при надсиланні або отриманні. Для даних такого типу потрібно спеціальна обробка та спеціальна взаємодія з ними, щоб уникнути більшості проблем, наприклад, використання застарілих даних або отримання даних, які проект вже отримав

раніше. Для даної задачі можна використовувати бібліотеку React Query, яка допомагає за допомогою хуків отримувати, зберігати та редагувати асинхронні дані та використання цих даних з будь-яких компонентів, не залежачи від того, де він знаходиться. [10]

Основні дві сутності, які будуть використані у даному проекті з React Query:

- `useQuery` — отримання даних, надавання цим даним унікального ключа, знаючи який пізніше можна оперувати цими даними з будь-якого місця в коді
- `useMutation` — редагування даних, де є схожий ключ на `useQuery`, який може поєднувати ці два хуки та допомагати автоматично оновлювати дані при певних діях, уникаючи непотрібних повторних запитів, які суттєво збільшують роботи та рендерів компонентів.

Поєднуючи React Redux та React Query можна чітко розділити синхронні та асинхронні дані у коді та використовувати їх як дві різні сутності, що є хорошим архітектурним рішенням, оскільки значно спрощує роботу з даними різних типів та суттєво зменшує кількість помилок, які утворюються як і при написанні коду, так і при невчасному запиті чи втраті даних через це. [11]



## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ

### 3.1 Поділ на основні компоненти

Проект поділений на декілька основний компонентів, кожен з яких відіграє свою роль та знаходиться за різними посиланнями:

- **Homepage** — домашня сторінка, на якій вказана вся загальна інформація про компанію, макретингові матеріали та інші дані. Посилання: <https://roas.pro/>. (Додаток 1)
- **VideoServices** — сторінка відеосервісів з описом того, як відбувається творення відеоматеріалів. Посилання: <https://roas.pro/services/video-marketing>. (Додаток 2)
- **Services** — сторінка сервісів, де описується створення всіх матеріалів, крім відео. Посилання: <https://roas.pro/services/social-media-video-production>. (Додаток 3)
- **Blog** — сторінка з навчальними та ознайомчими статтями. Посилання: <https://roas.pro/blog>. (Додаток 4)
- **Admin** — сторінка, на якій адмін має можливість вносити зміни в контент сайту та створювати нові статті <https://roas.pro/admin>.

### 3.2 Компонент Homepage

Даний компонент є найбільшим у проекті та описує загальні положення веб-ресурсу та можливості, які він надає. Також компонент Homepage повинен висвітлювати вже готові приклади маркетингових відеоресурсів та забезпечити зручний перехід до важливих інформаційних розділів ресурсу.

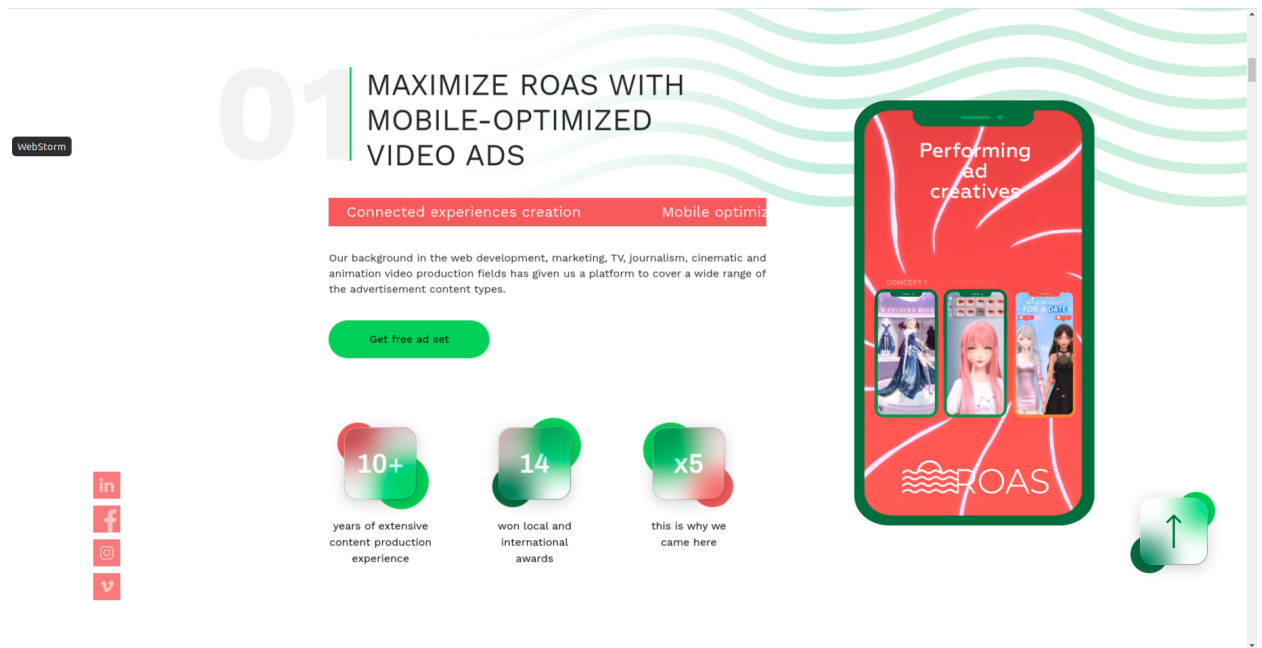


Рис 3.1. Частина компоненту Номерpage

Частини, на які поділений Номерpage:

- Ознайомчий компонент;
- Опис української компанії goas по створенню відеоресурсів;
- Готові кейси рекламних відео з описом;
- Сервіси, які надаються користувачам;
- Процес розробки матеріалів;
- Додаткові можливості компанії;
- Експертиза створених реклам;
- Плани розробки;
- Освітній та інформаційний блог;
- Питання та відповіді на них;
- Контакти.

### 3.3 Компоненти VideoServices та Services.

Компонент VideoServices описує все, що потрібно може бути потрібно користувачу, який прагне замовити або створити самостійно рекламний відеоролик (Рис 3.2).

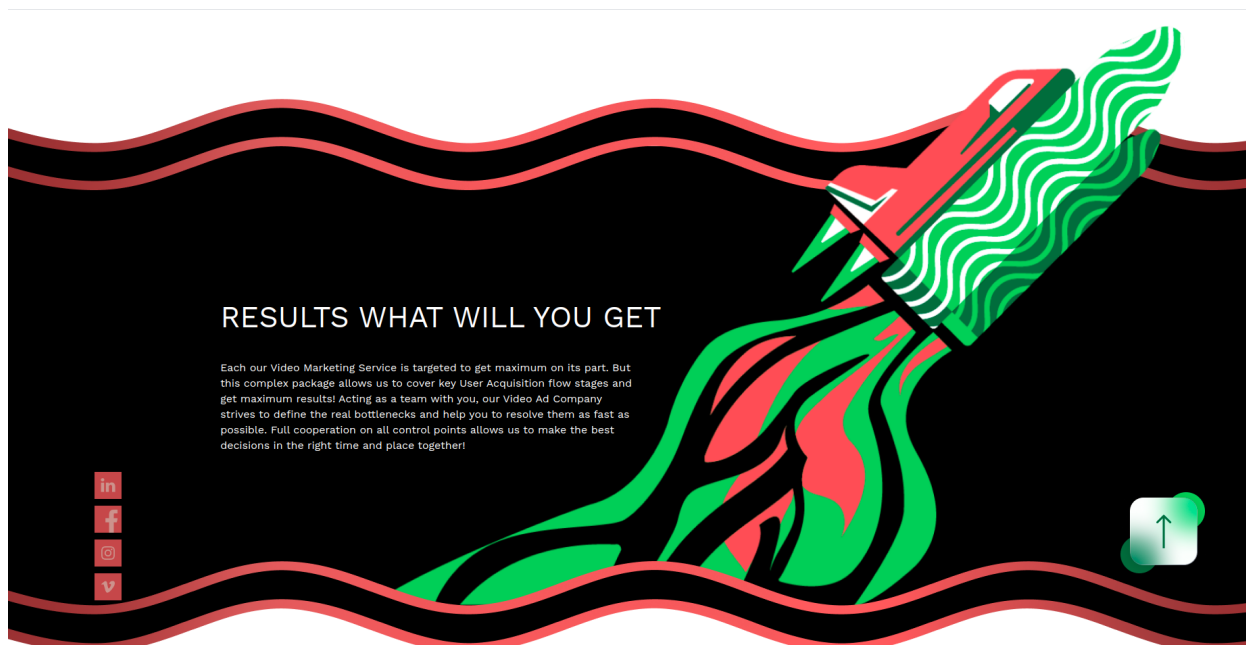


Рис 3.2. Частина компоненту VideoServices

Компонент Services в свою чергу описує все, що потрібно може бути потрібно користувачу, який прагне дізнатися щось про будь-які види рекламних продуктів, крім відеоматеріалів (Рис 3.3). Відповідно структура компонентів Services і VideoServices дуже наближена та має такий вигляд:

- Ознайомча сторінка;
- Пропозиції та поради;
- Схожі кейси;
- Результати;
- Персональна консультація;
- Інформаційний блог;

- Форма для контакту.



Рис 3.3. Частина компоненту Services

### 3.4 Компоненти Blog та Admin.

Компонент Blog займає дуже важливу роль та передбачає те, що користувач буде проводити на ньому найбільше часу. Тут описані всі інформаційні та освітні статті, які допомагають висвітлювати певні поширені проблеми зв'язані з маркетингом та відповідними продуктами (Рис 3.4).

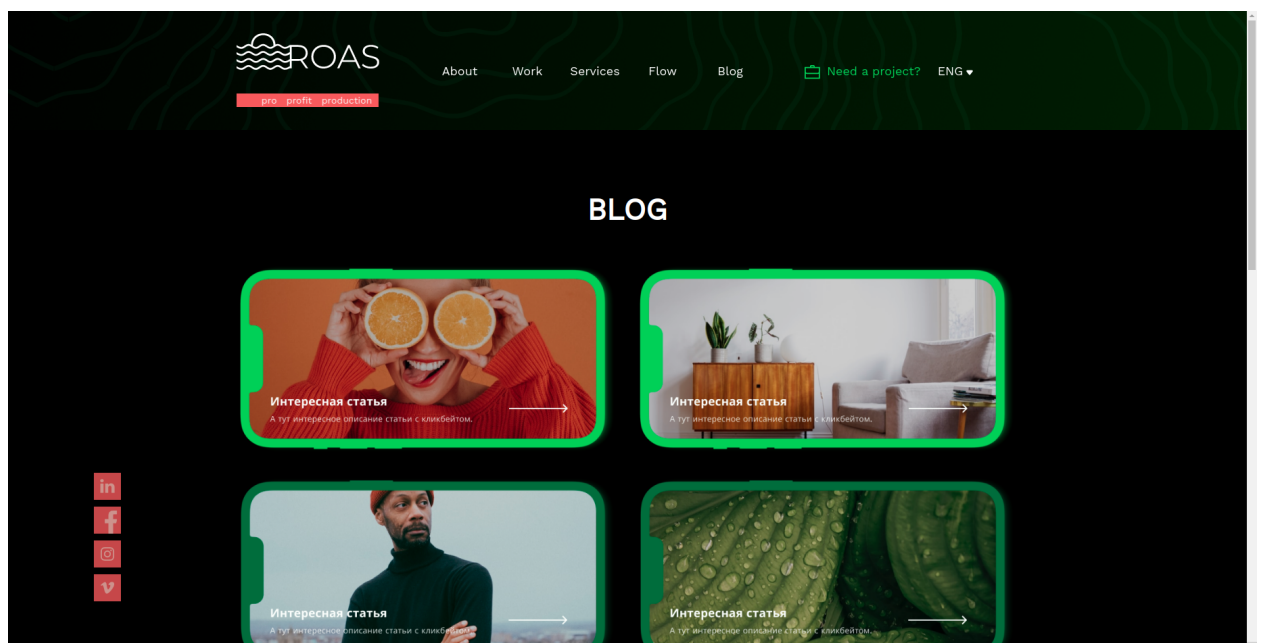


Рис 3.4. Компонент Blog

Користувач може обрати статтю, яка його зацікавила, та буде перенаправлений на інший URL, де інформація про неї буде отримана з бази даних та відображена користувачу.

Компонент Blog та Admin досить тісно пов'язані, оскільки всі статті адмін створює через адмін-сторінку та може взаємодіяти там з усіма компонентами, які присутні в компоненті Blog. Також адмін має змогу змінити не тільки вже створені статті, а й редагувати будь-який текст чи посилання на всіх сторінках веб-сайту.

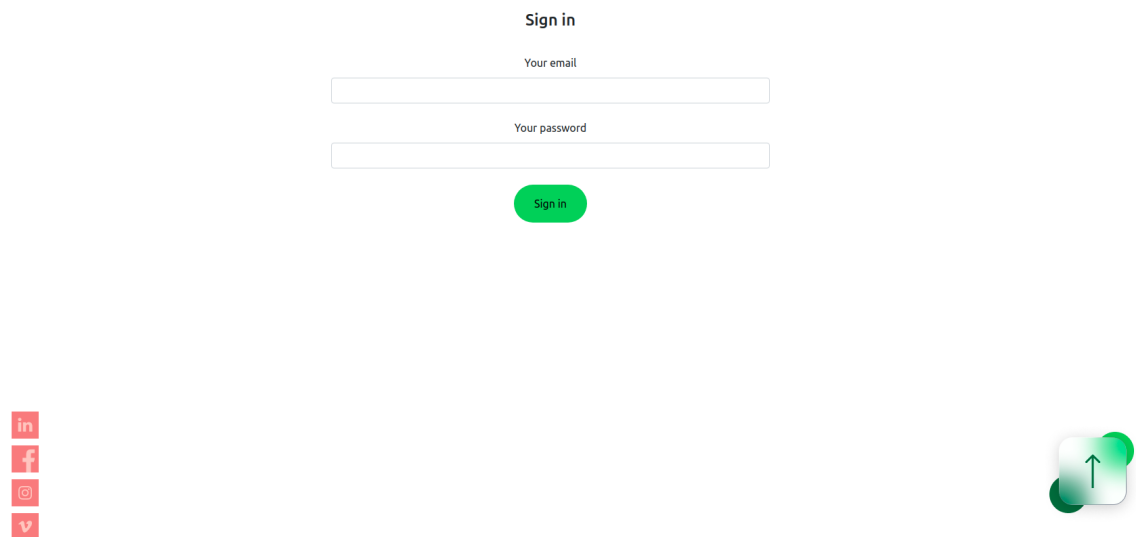
The image shows a login form for an admin interface. At the top, it says "Sign in". Below this are two input fields: "Your email" and "Your password". Under the password field is a green "Sign in" button. To the left of the form are four social media icons (LinkedIn, Facebook, Instagram, and YouTube) stacked vertically. To the right is a green square button with a white upward-pointing arrow.

Рис 3.5. Вхід в адмін-сторінку.

Створення статті через адмін-сторінку відбуває за допомогою додавання будь-якої кількості блоків та поділів кожної з них на клітини. Адмін може змінювати будь-які налаштування, такі як ширина клітини, тип контенту, висота, форматування і тд. Дані опції дозволяють створити максимально гнучкий інструмент по створенню різних статей.

## ВИСНОВКИ

Під час виконання роботи було розроблено веб-додаток, який дозволяє користувачам знаходити різну інформацію про нагальні потреби сучасних технік створення маркетингових матеріалів. З допомогою даного веб-додатку користувачі мають змогу спростити вивчення та покращити свої знання з впливу та розвитку реклами на бізнес на людей в цілому. Також для користувачів, які не мають намір вивчати дану тему, запропоновано замовити послугу для створення відеореклами його компанії чи ресурсу. Було проведено дослідження особливостей бібліотек та фреймворків, які найкраще візуалізують ці особливості, та обрані найзручніші і додано взаємодію з Google Firestore Database. Тому за даними вимогами був створений та опублікований додаток, який зараз можна побачити та дослідити за посиланням <https://roas.pro/>.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Video Advertising In Furute, - Режим доступу: <https://instapage.com/blog/video-future-of-advertising;>
- 2 2020 Global Digital Ad Trending, - Режим доступу: [https://pubmatic.com/reports/2020-global-ad-trends/;](https://pubmatic.com/reports/2020-global-ad-trends/)
- 3 ReactJs documentation, - Режим доступу: [https://uk.reactjs.org/;](https://uk.reactjs.org/)
- 4 React Redux documentation, - Режим доступу: [https://react-redux.js.org/;](https://react-redux.js.org/)
- 5 WebStorm IDE, - Режим доступу: <https://www.jetbrains.com/help/webstorm/viewing-javascript-reference.html>
- 6 Google Firebase documentation, - Режим доступу: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore;>
- 7 NoSQL databases, - Режим доступу: <https://www.mongodb.com/nosql-explained;>
- 8 React Redux store documentation, - Режим доступу: <https://redux.js.org/tutorials/fundamentals/part-2-concepts-data-flow;>
- 9 React Redux actions and reducers, - Режим доступу: <https://redux.js.org/tutorials/fundamentals/part-3-state-actions-reducers;>
- 10 React Query documentation, - Режим доступу: <https://react-query.tanstack.com/overview;>
- 11 React Redux and React Query connection, - Режим доступу: <https://react-query.tanstack.com/guides/does-this-replace-client-state;>