



Politechnika Łódzka

Instytut Informatyki

## PRACA DYPLOMOWA INŻYNIERSKA

# **Portfolio i blog z systemem komentowania dla prezentacji osiągnięć fotografa i dziennikarza muzycznego**

**Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej**

**Promotor: dr inż. Joanna Ochelska-Mierzejewska**

**Diplomant: Oskar Kurczewski**

**Nr albumu: 229935**

**Kierunek: Informatyka Stosowana**

**Specjalność: Infrastruktura i Aplikacje Sieciowe**

Łódź, luty 2023



Instytut Informatyki

90-924 Łódź, ul. Wólczańska 215, budynek B9

tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

# Spis treści

<b>1 Wstęp</b>	<b>4</b>
1.1 Cel i założenia pracy . . . . .	6
1.2 Układ pracy . . . . .	6
<b>2 Przegląd istniejących rozwiązań</b>	<b>7</b>
2.1 Personalne strony-wizytówki fotografów . . . . .	7
2.1.1 Ashley Osborn . . . . .	7
2.1.2 Zackery Michael . . . . .	12
2.2 Portale społecznościowe . . . . .	13
2.2.1 Instagram . . . . .	14
2.2.2 Facebook . . . . .	16
2.3 Podsumowanie przeglądu . . . . .	19
<b>3 Narzędzia wybrane do realizacji projektu</b>	<b>20</b>
<b>4 Projekt portfolio i blogu</b>	<b>26</b>
4.1 Założenia projektu . . . . .	26
4.2 Wymagania funkcjonalne . . . . .	26
4.3 Wymagania niefunkcjonalne . . . . .	30
4.4 Realizacja projektu . . . . .	31
4.4.1 Strapi — stworzenie struktury danych oraz konfiguracja systemu zarządzania treścią . . . . .	31
4.4.2 Next.js — implementacja strony internetowej . . . . .	37
<b>5 Portfolio i blog</b>	<b>45</b>
5.1 System zarządzania treścią . . . . .	45
5.2 Strona internetowa . . . . .	48

<b>6 Podsumowanie</b>	<b>49</b>
<b>Spis rysunków</b>	<b>50</b>
<b>Spis tabel</b>	<b>52</b>
<b>Spis listingów</b>	<b>53</b>
<b>Bibliografia</b>	<b>54</b>

# 1 Wstęp

Fotografia to stosunkowo nowy wynalazek — nie minęło nawet 200 lat od wykonania pierwszego znanego nam zdjęcia, słynnego “Widoku z okna w Le Gras” z 1826 roku [1]. Początki uwieczniania obrazu przez ludzi były jednak diametralnie różne od tego, co znane jest dzisiaj. Do momentu wynalezienia matrycy światłoczułej (czyli powstania fotografii cyfrowej), rozwój technik fotograficznych polegał jedynie na ulepszaniu technologii bazującej na światłoczułości soli srebra [1]. Fotografia wkrótce została rozpowszechniona na całym świecie. Dało jej to ogromne znaczenie historyczne i kulturowe — zdjęcia mogą być zarówno dziełami sztuki, jak i dokumentacją ważnych wydarzeń. Powstał również zawód fotografa, który przeszedł wiele zmian na przestrzeni czasu — początkowo były to osoby, do których należało się udać, żeby mieć możliwość uwiecznienia swojego wizerunku. Zmiana charakterystyki wspomnianej profesji zaczęła następować wraz z wynalezieniem fotografii cyfrowej oraz rozwojem technologii informatycznych. W 1957 roku powstał pierwszy obraz cyfrowy, wykonany przez Russella Kirscha, amerykańskiego naukowca-informatyka. Był to skan analogowej fotografii jego 3-miesięcznego syna [3]. Dwanaście lat później Willard S. Boyle oraz George E. Smith wynaleźli matrycę CCD<sup>1</sup>. Z perspektywy czasu okazało się to być na tyle istotnym dokonaniem, że jego autorzy otrzymali w 2009 roku nagrodę Nobla [2]. Powstanie i rozpowszechnienie fotografii cyfrowej zbiegło się też z okresem najszybszego wzrostu ogólnej dostępności Internetu. Na przełomie tysiącleci pojawiły się media społecznościowe, w tym niektóre niemal całkowicie oparte na publikowaniu obrazów, na przykład Instagram ([www.instagram.com](http://www.instagram.com)). Ludzie codziennie dzielą się milionami zdjęć, a największe konta posiadają nawet kilkaset milionów obserwujących [4]. Inne media społecznościowe również w dużej mierze polegają na możliwości udostępniania zdjęć. Wszystko to powoduje, że zawód fotografa jest obecnie zupełnie inny, niż

---

<sup>1</sup>ang. CCD — charge-coupled device.

w czasach przed wynalezieniem aparatu cyfrowego. Pojawiają się nowe rodzaje zleceń — profesjonalnie wykonane zdjęcia potrzebne są nie tylko po to, aby uwiecznić przełomowe momenty w życiu człowieka lub umieścić je w prasie, ale też stanowią niezwykle ważny środek przekazu w reklamie.

Powszechna dostępność aparatów cyfrowych oraz rozwój technologii pozwalający na polepszenie stosunku ceny do jakości sprzętu fotograficznego spowodowały jednak, że pojawiło się wielu fotografów, a co za tym idzie — utworzyła się większa konkurencja na rynku. Trudniej jest znaleźć nowych klientów. Aby pozyskać zlecenia, w obecnych czasach konieczna jest więc odpowiednia promocja świadczonych usług. W 2022 roku kanadyjski magazyn Format przeprowadził ankietę. Brało w niej udział 3 898 fotografów z 97 krajów. Według zebranych odpowiedzi, na najważniejsze źródła pozyskiwania nowych klientów składają się polecenia od poprzednich klientów (61% respondentów), strona internetowa z portfolio (40%), Instagram (38%), Facebook (25%) oraz wyszukiwania w Google (22%) [7]. Z danych tych wynika, że najbardziej istotnym internetowym narzędziem do pozyskiwania nowych klientów są personalne strony internetowe z portfolio fotografa.

To, jak ma wyglądać portfolio osoby świadczącej usługi fotograficzne, w dużym stopniu zależy od tego, jaką dziedzinę fotografii reprezentuje jego właściciel. Przykładowo, fotograf ślubny oprócz samego portfolio powinien posiadać na swojej stronie internetowej dokładny opis świadczonych usług, a także przydatne informacje pomagające potencjalnemu klientowi podjąć decyzję o kontakcie. Osobie prowadzącej studio fotograficzne przyda się wykaz rekwizytów, a fotografowi produktowemu lista firm, z którymi współpracował. Strona tworzona w ramach pracy inżynierskiej będzie natomiast przeznaczona dla fotografa koncertowego i dziennikarza muzycznego. Przydatnym elementem będzie więc blog, który posłuży do umieszczania relacji z wydarzeń takich, jak koncerty czy festiwale, a także zapewni miejsce do publikowania wypowiedzi w charakterze felietonu lub recenzji.

## **1.1 Cel i założenia pracy**

Celem pracy jest zaprojektowanie oraz implementacja personalnej strony internetowej łączącej cechy strony-wizytówki, blogu z systemem komentowania oraz portfolio fotograficznego. Strona ta ma być miejscem prezentacji osiągnięć fotografa i dziennikarza muzycznego. Etap projektu skupia się na przeglądzie istniejących rozwiązań oraz na wykonaniu projektu graficznego strony. Implementacja polega na stworzeniu dwuczęściowej aplikacji sieciowej przy użyciu systemu zarządzania treścią Strapi oraz szkieletu aplikacyjnego Next.js, umożliwiającego pobieranie danych ze Strapi oraz renderowanie treści po stronie serwera, co pozwala na korzystne pozycjonowanie strony w wyszukiwarkach internetowych.

## **1.2 Układ pracy**

Rozdział pierwszy wprowadza do tematyki oraz problematyki pracy. Drugi rozdział jest przeglądem istniejących rozwiązań — personalnych stron-wizytówek popularnych zagranicznych fotografów koncertowych, portali społecznościowych używanych w branży, a także blogów i innych stron o tematyce muzycznej. W rozdziale trzecim przedstawione są narzędzia, które zostały wykorzystane podczas projektowania oraz implementacji tworzonej strony internetowej. Rozdział czwarty poświęcony jest projektowi portfolio i blogu — znalazły się w nim założenia projektu, wymagania funkcjonalne i niefunkcjonalne oraz opis realizacji projektu. Piąty rozdział jest prezentacją wykonanej strony internetowej, a szósty podsumowuje realizację projektu, celu oraz założeń pracy.

## 2 Przegląd istniejących rozwiązań

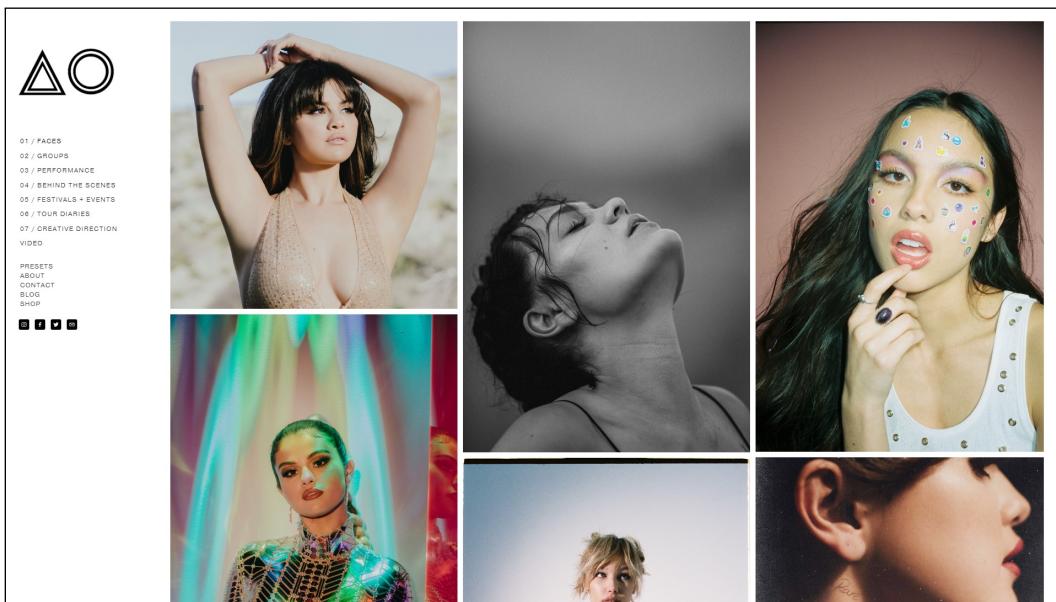
Fotografia i dziennikarstwo to branże, które wzajemnie od siebie zależą. Zdjęcia urozmaicają relacje prasowe z wydarzeń, powodując, że artykuły są ciekawsze, a odbiór fotoreportażu jest lepszy wtedy, gdy przedstawiona jest jakaś historia. Aby strona internetowa realizowana na potrzeby tej pracy inżynierskiej była wykonana z zachowaniem najlepszych możliwych praktyk, został przeprowadzony przegląd istniejących rozwiązań posiadających podobne funkcje. Pozwoli to na znalezienie inspiracji, wyciągnięcie wniosków i wykorzystanie ich zarówno podczas projektowania wyglądu strony, jak i na etapie implementacji.

### 2.1 Personalne strony-wizytówki fotografów

Personalne strony-wizytówki fotografów są zazwyczaj nieskomplikowane pod względem budowy, nawigacji oraz wyglądu. Rozkład elementów powinien być możliwie prosty, żeby potencjalny klient oglądający portfolio mógł skupić się na jego zawartości. W tym podrozdziale zostaną poddane analizie dwie strony profesjonalnych fotografów działających w branży muzycznej.

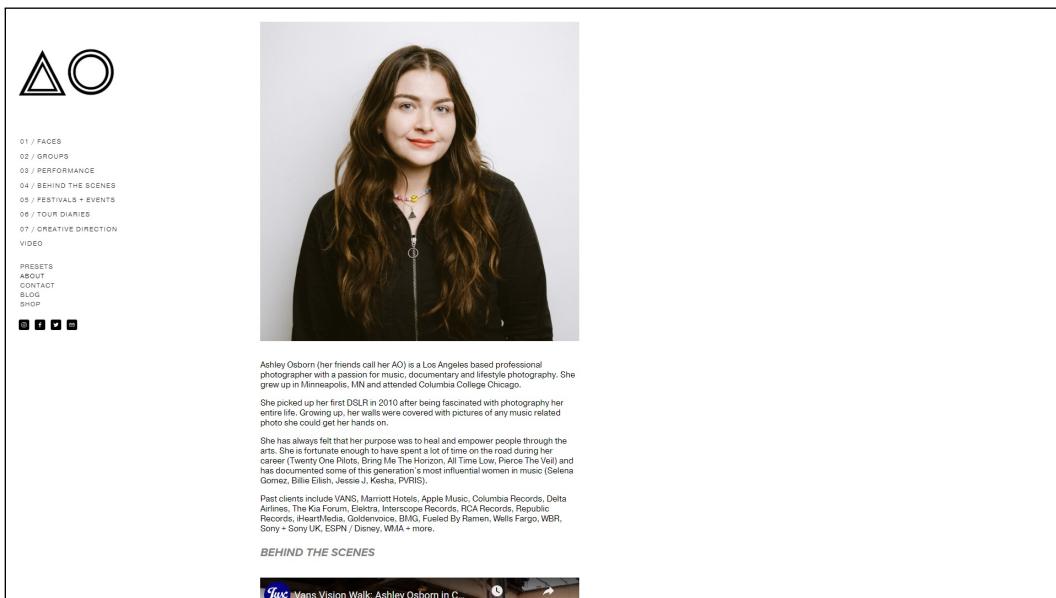
#### 2.1.1 Ashley Osborn

Ashley Osborn jest amerykańską fotografką zamieszkałą w Los Angeles. Specjalizuje się w dokumentowaniu tras koncertowych, ale zajmuje się też wykonywaniem sesji zdjęciowych oraz tworzeniem teledysków. Współpracowała z najpopularniejszymi artystami muzycznymi z całego świata oraz globalnymi firmami z branży kulturowej [6].



Rys. 2.1: [www.ashleyosborn.com](http://www.ashleyosborn.com) — strona główna.

Rysunek 2.1 przedstawia wygląd strony głównej witryny internetowej [www.ashleyosborn.com](http://www.ashleyosborn.com). Warto zwrócić szczególną uwagę na kilka jej elementów. W lewym górnym rogu znajduje się logo będące inicjałami właścicielki analizowanej strony. Pełni ono funkcję estetyczną i jest ważną częścią identyfikacji wizualnej. Ponadto, będąc w dowolnym miejscu na stronie fotografki, można na nie kliknąć — spowoduje to przeniesienie użytkownika na stronę główną. Oprócz logo, w lewej części strony umieszczone zostało menu, które też jest zawsze widoczne. Resztę, a zarazem większość ekranu, wypełniają zdjęcia. Warto zaznaczyć, że ich wybór nie jest przypadkowy — osoby, które się na nich pojawiły, są znanyymi na świecie artystkami. Przykładowo osoba pozująca na pierwszej fotografii od lewej to Selena Gomez - posiada ona największą liczbę (ponad 373 miliony) obserwujących na Instagramie [4].



Rys. 2.2: [www.ashleyosborn.com](http://www.ashleyosborn.com) — sekcja “o mnie”.

The screenshot shows a contact form on the website. At the top left is the logo 'ΔO'. Below it is a message: 'Hi, feel free to use the form below to get in touch or email me directly at [a@ashleyosborn.com](mailto:a@ashleyosborn.com)'. The form has several input fields: 'Name \*' (split into 'FirstName' and 'LastName'), 'Email Address \*', 'Subject \*', and 'Message \*'. At the bottom of the form is a 'SUBMIT' button.

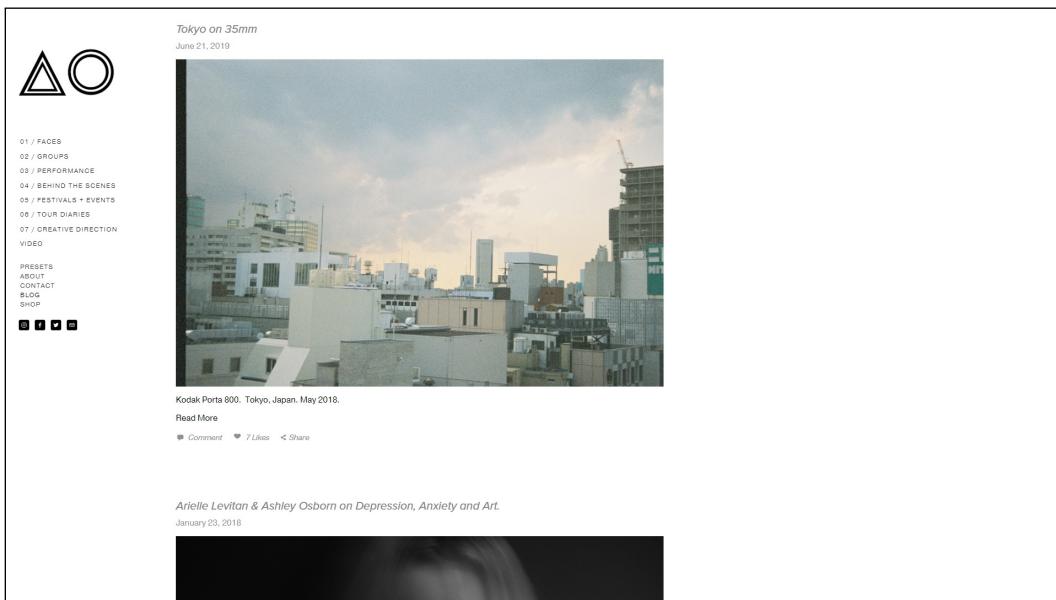
Rys. 2.3: [www.ashleyosborn.com](http://www.ashleyosborn.com) — formularz kontaktowy.

Widoczna na rysunku 2.2 sekcja “o mnie” jest kolejnym nieodłącznym elementem każdej strony-wizytówki należącej do osoby zajmującej się profesjonalną fotografią. Pozwala ona przedstawić wygląd, charakter i osiągnięcia osoby, która chce zachęcić potencjalnych klientów do skorzystania z usługi. Nie musi być ona rozbudowana — w tym przypadku są to portret oraz krótki, lecz treściwy opis życiorysu Ashley Osborn. Zawiera on między innymi wykaz najważniejszych zleceń z przeszłości, nazwę ukończonej szkoły wyższej, a także obecne miejsce zamieszkania. Formularz kontaktowy z rysunku 2.3 umożliwia szybkie wysłanie wiadomości bez konieczności używania poczty elektronicznej lub mediów społecznościowych. Warto zwrócić uwagę na gwiazdki przy polach obowiązkowych (w tym przypadku — po prostu wszystkich polach). Wprowadzane dane powinny być walidowane, a użytkownik poinformowany zarówno wtedy, gdy próba wysłania danych powiedzie się, jak i wtedy, gdy wystąpi błąd [9].

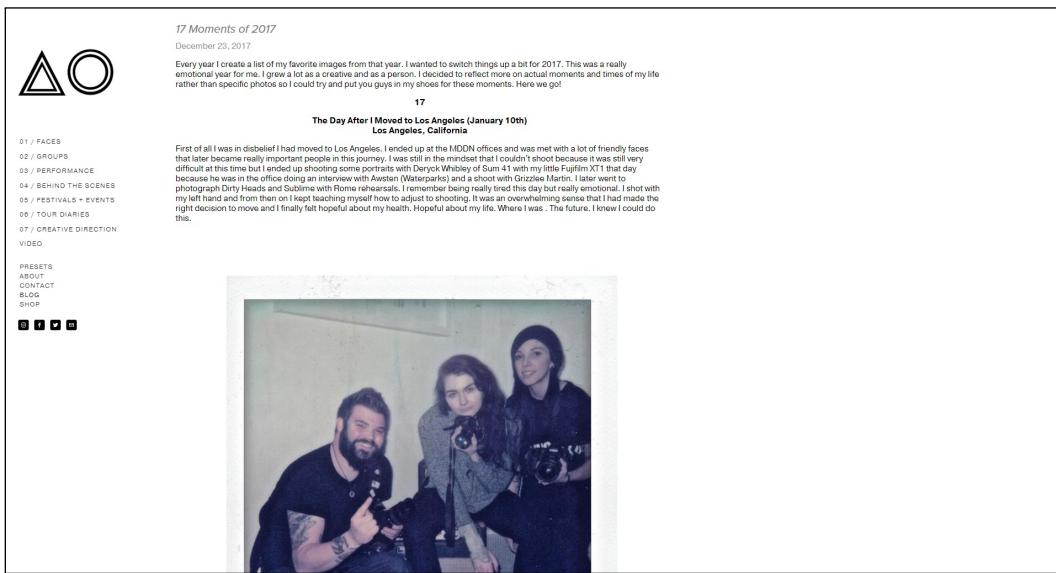
Ostatnią sekcją strony Ashley Osborn jest blog, skonstruowany w prosty sposób, podobny do reszty strony. Użytkownik może wybrać interesujący go post z listy (rys. 2.4), po czym przenoszony jest do jego treści (rys. 2.5). Należy tu się przyjrzeć tekstowi — nie zajmuje on całej szerokości strony. Ułatwia to czytanie — zarówno zbyt szerokie, jak i zbyt wąskie akapity skutecznie zniechęcają użytkownika do czytania [10]. Oprócz rzeczy przedstawionych na powyższych zrzutach ekranu, na stronie istnieje też możliwość przejścia do sklepu, w którym sprzedawane są *presety*<sup>2</sup> do programu do obróbki zdjęć Adobe Lightroom.

---

<sup>2</sup>ang. preset — zestaw gotowych ustawień, najczęściej wykorzystywany przy obróbce dużej liczby zdjęć, pozwalający na utrzymanie spójnej stylistyki na każdym z nich.

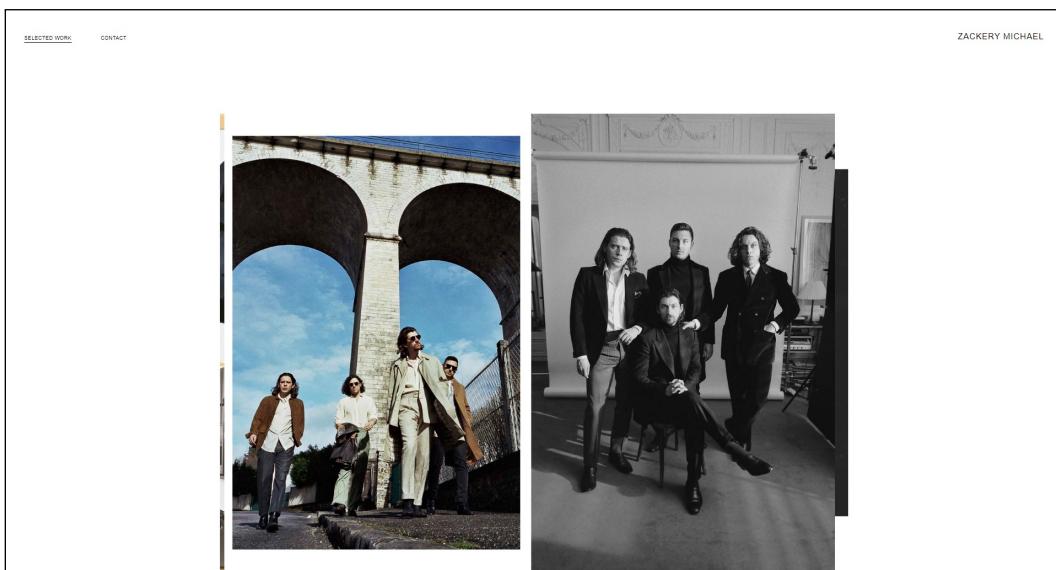


Rys. 2.4: [www.ashleyosborn.com](http://www.ashleyosborn.com) — blog (lista postów).



Rys. 2.5: [www.ashleyosborn.com](http://www.ashleyosborn.com) — blog (widok postu).

## 2.1.2 Zackery Michael



Rys. 2.6: [www.zackerymichaelstudio.com](http://www.zackerymichaelstudio.com) — strona główna.



Rys. 2.7: [www.zackerymichaelstudio.com](http://www.zackerymichaelstudio.com) — sekcja kontaktowa.

Druga przedstawiona strona-wizytówka jest zupełnie inna w porównaniu do poprzedniej. O ile sam wygląd pierwszej analizowanej witryny był prosty, to ilość treści była dość bogata (blog, sekcja “o mnie”, oddzielny formularz kontaktowy czy nawet sklep). W przypadku portfolio nowojorskiego fotografa Zackery’ego Michaela zastosowano zdecydowanie bardziej minimalistyczne podejście. Jego witryna internetowa składa się jedynie z dwóch części: strony głównej, przeznaczonej do zaprezentowania wybranych prac (rys. 2.6) oraz sekcji “kontakt”, gdzie obok zdjęcia przedstawiającego Zackery’ego Michaela podany jest jego adres poczty elektronicznej oraz przedstawione są najważniejsze firmy, z którymi współpracował fotograf (rys. 2.7). Tekst jest znacznie ograniczony w celu zwrócenia pełnej uwagi odwiedzających stronę na prace zaprezentowane przez Zackery’ego Michaela.

## 2.2 Portale społecznościowe

Wraz z rozwojem technologii internetowych pojawiły się portale społecznościowe, które w dzisiejszych czasach pełnią bardzo ważną funkcję zarówno w znaczeniu społecznym (ułatwiają kontakt oraz zawieranie nowych znajomości przez Internet), jak i biznesowym (ze względu na wielką popularność są potężnym narzędziem marketingowym i źródłem pozyskiwania klientów). W kontekście zawodu fotografa są one o tyle ważne, że Instagram oraz Facebook, czyli te najbardziej popularne, są w dużej mierze oparte na udostępnianiu zdjęć. Pomaga to w budowaniu marki osobistej osoby świadczącej usługi fotograficzne, co ułatwia zdobywanie nowych zleceń i poszerzanie działalności poza rynek lokalny [11]. Wspomniane portale społecznościowe zostaną w tym rozdziale opisane pod kątem prezentacji treści, zwłaszcza postów (Facebook) i zdjęć (Instagram) oraz profilów użytkownika.

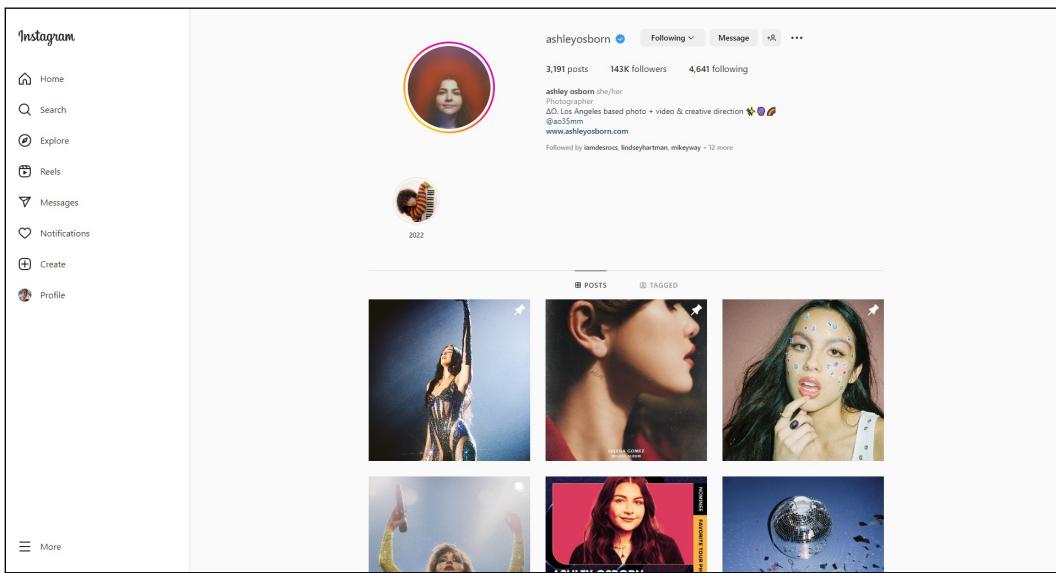
### 2.2.1 Instagram

Instagram to powstały w 2010 roku [16] portal społecznościowy skupiony na możliwości udostępniania zdjęć i filmów, które następnie mogą być polubione<sup>3</sup> oraz komentowane przez innych użytkowników. Wśród fotografów jego popularność jest szczególnie duża ze względu na możliwość pełnienia przez ten serwis funkcji portfolio oraz łatwość podjęcia kontaktu poprzez wiadomości bezpośrednie. Przykładowym zaprezentowanym profilem użytkownika będzie widoczny na rysunku 2.8 profil Ashley Osborn, fotografki, której strona internetowa została opisana w podrozdziale 2.1.1. Składa się on z kilku elementów, którymi są:

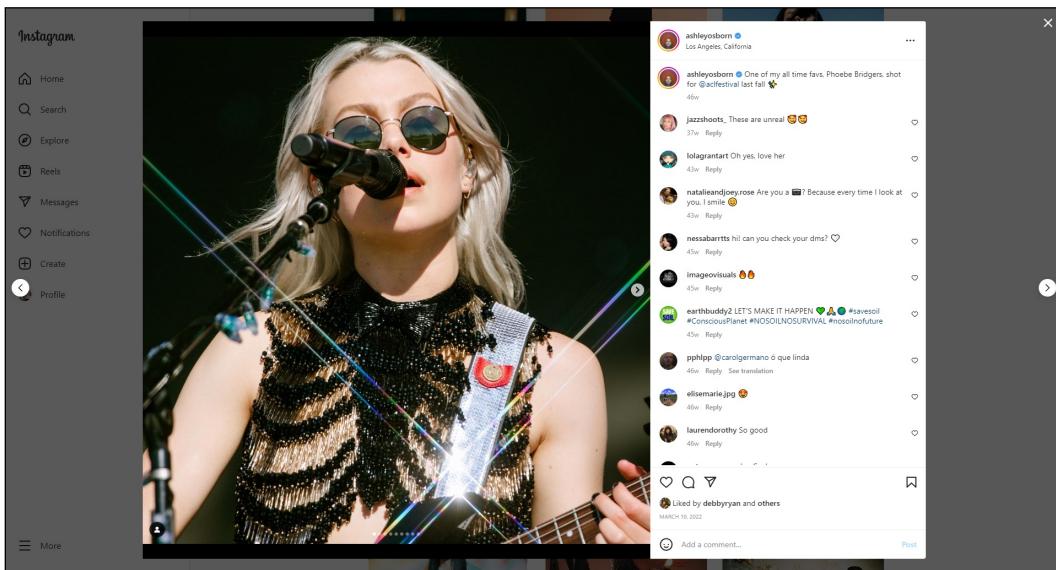
- Część nagłówkowa ze zdjęciem profilowym, informacjami o liczbie postów, osób obserwujących oraz obserwowanych, opisem profilu oraz linkiem do strony internetowej.
- Zapisane relacje (ang. *stories* — wprowadzony w 2016 roku mechanizm Instagrama, pozwalający na udostępnianie zdjęć oraz filmów znikających po 24 godzinach [15]).
- Galeria w postaci szerokiej na trzy elementy siatki kwadratowych zdjęć lub filmów. Warto zwrócić uwagę, że trzy pierwsze posty są przypięte, a kolejne — ułożone w kolejności od najnowszych do najstarszych. Opcja przypinania postów daje użytkownikom większy wpływ na wygląd ich profilu poprzez wyróżnienie ulubionych zdjęć lub filmów.

---

<sup>3</sup>polubienie, potocznie lajk (ang. like) — wyrażenie aprobaty wobec treści poprzez np. kliknięcie kciuka w górę lub symbolu serca.



Rys. 2.8: Instagram — profil użytkownika.



Rys. 2.9: Instagram — widok postu.

Kolejnym widokiem jest widok postu (rys. 2.9). Pozwala on na przeglądanie wszystkich znajdujących się w nim filmów i zdjęć, czytanie i dodawanie komentatorzy, polubienie postu oraz udostępnienie go za pomocą wiadomości prywatnej lub

skopiowanie linku, aby umieścić go poza Instagramem. Warto zwrócić uwagę na przyciemnioną pozostałą część strony — użytkownik po zamknięciu postu wróci do miejsca, od którego zaczął jego oglądanie.

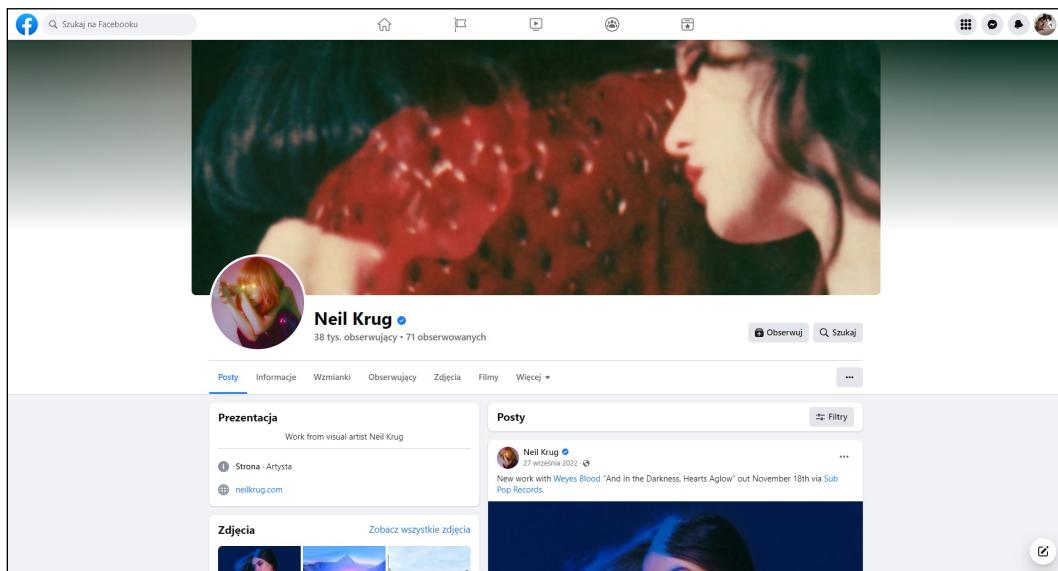
### 2.2.2 Facebook

Założony w 2004 roku [14] Facebook to najpopularniejszy pod względem miesięcznej liczby użytkowników serwis społecznościowy [5]. Portal ten jest wykorzystywany zarówno do tworzenia profilów prywatnych, jak i firmowych. Użytkownicy mogą dodawać się nawzajem do grona znajomych, umieszczać na swojej tablicy posty tekstowe lub ze zdjęciami i filmami, tworzyć grupy, obserwować interesujące ich strony. Te ostatnie mogą reprezentować np. popularnych artystów, lokalne firmy lub po prostu umieszczać treści o określonej tematyce. Ze względu na ogólną popularność i idącym za nią potencjałem marketingowym posty publikowane przez strony mogą też być promowane, co zwiększa potencjalną liczbę odbiorców. Elementami profilu użytkownika są:

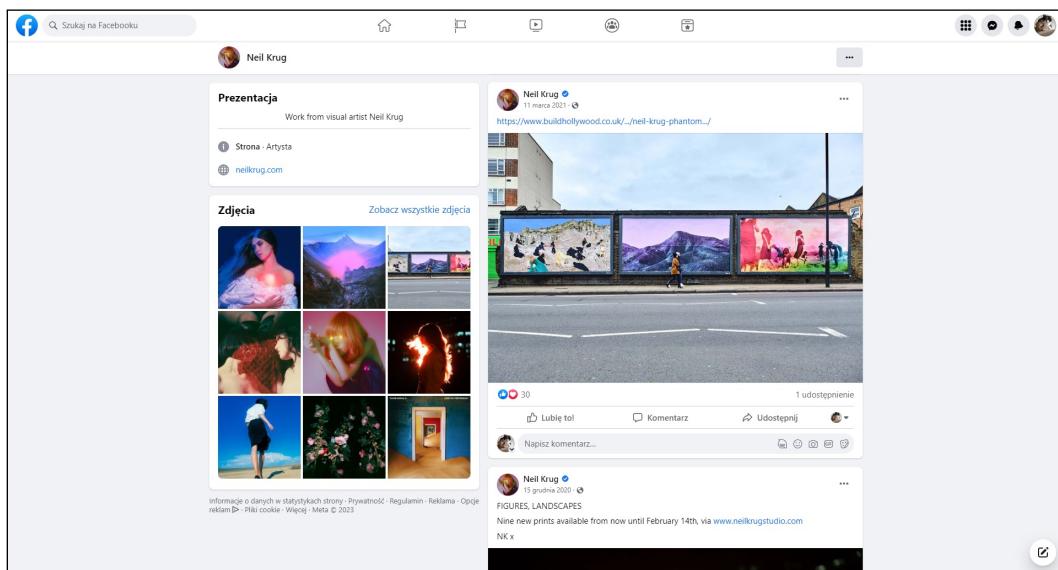
- część nagłówkowa zawierająca tło, zdjęcie profilowe, nazwę strony, liczbę osób obserwujących i obserwowanych, możliwość zaobserwowania profilu oraz opcję wyszukiwania (rys. 2.10);
- sekcja o nazwie “Prezentacja” — krótki opis, informacja o rodzaju strony oraz odniesienie do strony internetowej (rys. 2.10);
- galeria zdjęć (rys. 2.11);
- uporządkowana chronologicznie (od najnowszego do najstarszego) lista postów (rys. 2.11).

Posty mogą być wyświetlane oddzielnie. W zależności od tego, czy razem z nimi dodane jest zdjęcie lub film, widok ten się różni. Rysunek 2.12 przedstawia post

ze zdjęciem, natomiast rysunek 2.13 — post bez zdjęcia z osadzonym linkiem. Podobnie jak w przypadku strony prezentowanej w podrozdziale 2.1.1, warto zwrócić uwagę na ograniczoną szerokość posta, a co za tym idzie — tekstu.



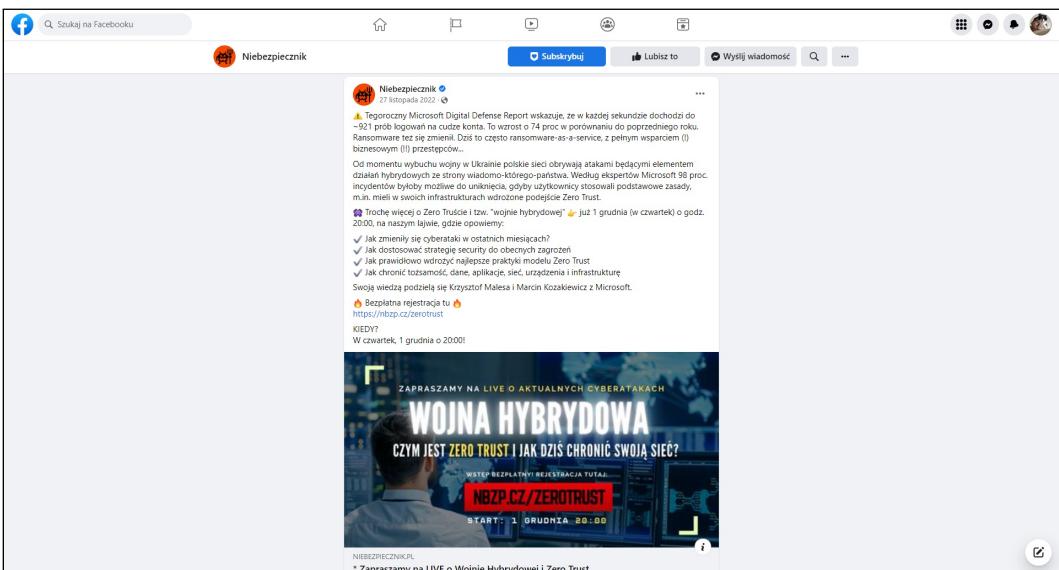
Rys. 2.10: Facebook — widok profilu użytkownika ze zdjęciem w tle.



Rys. 2.11: Facebook — widok tablicy użytkownika po przewinięciu.



Rys. 2.12: Facebook — widok postu ze zdjęciem.



Rys. 2.13: Facebook — widok postu tekstowego (z osadzonym linkiem).

## **2.3 Podsumowanie przeglądu**

Biorąc pod uwagę cechy analizowanych stron-wizytówek oraz portali społecznościowych używanych do pozyskiwania nowych klientów i promowania własnych prac, wyciągnięte zostały wnioski dotyczące podejścia do tworzenia projektu strony internetowej oraz planowania jej funkcjonalności. Po pierwsze, strony internetowe zawierające portfolio powinny być proste pod względem nawigacji (np. widoczny cały czas pasek nawigacyjny) oraz projektu graficznego (minimalistyczny układ i prosta kolorystyka pozwalają skupić się na treści). Po drugie, w przypadku występowania większej ilości tekstu, wskazane jest zachowanie ograniczonej szerokości bloku tekstu. Strona główna jest pierwszym miejscem, w które trafia użytkownik, więc przydadzą się na niej skupiające uwagę zdjęcia (może to być spektakularna sceneria, ale też na przykład ktoś sławny). Sekcja “o mnie” powinna natomiast zawierać zwięzłe opisany życiorys oraz doświadczenie zawodowe. W przypadku obecności formularza kontaktowego należy pamiętać o validacji wprowadzanych przez użytkownika danych oraz wyświetlaniu rezultatu próby wysłania danych. W czasach ogromnej popularności mediów społecznościowych dobrą praktyką jest też umieszczenie odnośników do znajdujących się tam profili fotografa.

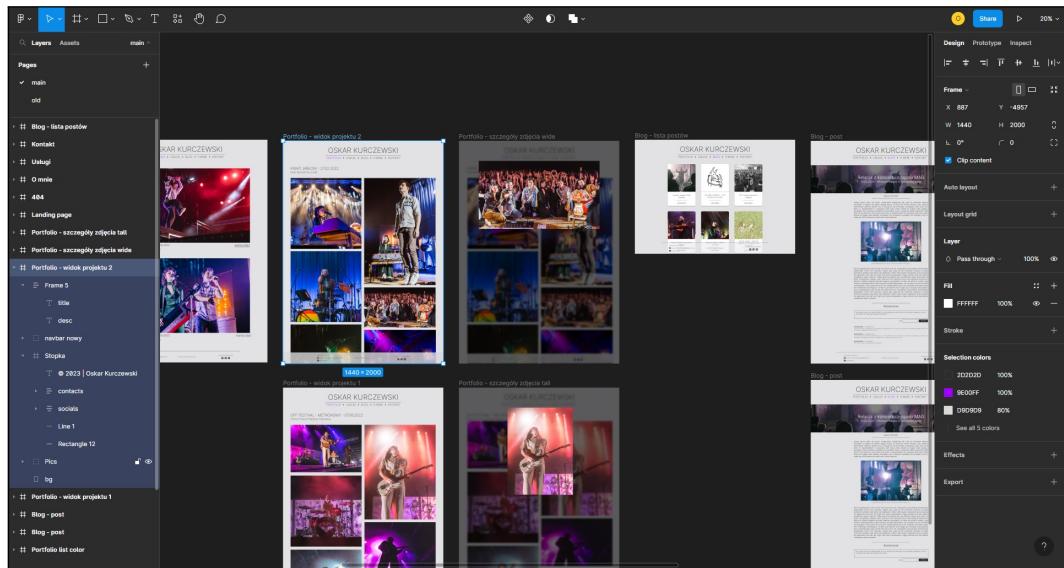
### 3 Narzędzia wybrane do realizacji projektu

Niniejszy rozdział przedstawia narzędzia, które zostały wykorzystane do wykonania projektu graficznego tworzonej strony oraz jego realizacji. Ich wybór podkutowany był wcześniejszym doświadczeniem, dostępnością dokumentacji oraz dostosowaniem do wymagań projektowych, które zostały przedstawione w rozdziale 4.

#### Figma



Rys. 3.1: Logo narzędzia Figma.



Rys. 3.2: Interfejs użytkownika narzędzia Figma.

Wcześniejsze wykonanie projektu graficznego tworzonej strony internetowej pozwala na jej łatwiejszą implementację. Taki interfejs użytkownika jest też bar-

dziej spójny wizualnie, co jest ważne dla pozytywnego odbioru strony. Do sporządzania projektu strony-portfolio z blogiem wykorzystane zostało narzędzie Figma (rys. 3.2) — prosty w obsłudze program pozwalający na wykonanie interaktywnych projektów interfejsów użytkownika za pomocą grafiki wektorowej [12].

## JavaScript



Rys. 3.3: Logo języka JavaScript.

Przy implementacji tworzonego projektu wykorzystany został JavaScript (JS), czyli najpopularniejszy język programowania na świecie [13]. Cechują go: wieloplatformowość, wysokopoziomowość i dynamiczne typowanie. Jako zaletę języka JavaScript można wymienić wsparcie dla wielu platform — tworzone za jego pomocą aplikacje mogą być używane zarówno na urządzeniach mobilnych, jak i w przeglądarce. Ponadto duży wybór bibliotek oraz szkieletów aplikacyjnych ułatwia dobranie odpowiednich narzędzi do danego typu projektu.

## Node.js oraz npm



Rys. 3.4: Logotypy Node.js oraz npm.

Node.js to wieloplatformowe środowisko uruchomieniowe zbudowane w języku JavaScript [19]. Jego działanie jest oparte na zdarzeniach asynchronicznych, dzięki

czemu aplikacja stworzona w Node.js działa jako jeden proces. Integralnym elementem środowiska Node.js jest npm (*Node Package Manager*) — system zarządzania pakietami pozwalający na łatwą instalację wybranych bibliotek i narzędzi [20].

## Strapi



Rys. 3.5: Logo systemu zarządzania treścią Strapi.

Strapi to system zarządzania treścią (CMS) typu *headless* — oznacza to, że nie pozwala on na definiowanie układu i wyglądu strony, a jedynie za zarządzanie treścią i jej dostarczanie do warstwy widoku. Użytkownik za pomocą intuicyjnego interfejsu graficznego może stworzyć model danych dostosowany do aktualnych potrzeb i wymagań. Do dyspozycji są dostępne typy pojedyncze (występujące tylko raz na całej stronie, np. treść sekcji “o mnie”), kolekcje (np. posty lub projekty w portfolio) oraz komponenty (elementy powtarzające się w kolekcjach i typach pojedynczych) [18]. Głównymi zaletami Strapi są:

- wsparcie dla wielu systemów zarządzania bazami danych — SQLite (domyślny), MySQL, PostgreSQL oraz MariaDB;
- wiele dostępnych wtyczek ułatwiających korzystanie z systemu;
- konfigurowalne API, pozwalające na dostosowanie systemu do potrzeb;
- obsługa kont użytkownika i uprawnień poprzez role, dzięki czemu nie ma konieczności jej oddzielnego implementowania.

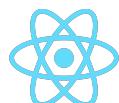
## SQLite



Rys. 3.6: Logo systemu zarządzania bazami danych SQLite.

SQLite jest domyślnym systemem zarządzania relacyjną bazą danych dla Strapi. W porównaniu do innych popularnych systemów takich jak MySQL czy PostgresQL, SQLite wyróżnia się tym, że jest biblioteką języka C i nie wymaga oddzielnego procesu — zapis i odczyt wykonywany jest po prostu za pomocą pliku bazy danych [21]. Dzięki temu jego ogromną zaletą są brak konieczności instalacji oraz konfiguracji bazy danych, a co za tym idzie — łatwość użytkowania. SQLite wspiera język zapytań SQL (ang. *Structured Query Language*).

## React



Rys. 3.7: Logo biblioteki React.

React to biblioteka będąca jednym z najpopularniejszych narzędzi do tworzenia interfejsów użytkownika stron internetowych [22]. Całość kodu pisana jest w JavaScript. Ważnym elementem Reacta jest JSX (ang. *JavaScript Syntax Extension*) — rozszerzenie składni JavaScript podobne do języka znaczników HTML. Ułatwia ono intuicyjne tworzenie komponentów, z których składa się strona internetowa.

## Next.js



Rys. 3.8: Logo szkieletu aplikacyjnego Next.js.

Next.js jest szkieletem aplikacyjnym dla biblioteki React pozwalającym na tworzenie statycznych aplikacji internetowych [23]. Jego kluczową cechą jest możliwość renderowania i generowania zawartości strony internetowej po stronie serwera, co ulepsza pozycjonowanie witryny w wyszukiwarkach (SEO — ang. *search engine optimization*). Dla ułatwienia pracy nad tworzonymi stronami internetowymi, Next.js korzysta z routingu opartego na stronach. Dodatkowo, wsparcie dla routingu dynamicznego zapewnia m. in. możliwość łatwego tworzenia czytelnych dla użytkownika linków w formie unikalnego ciągu znaków (ang. *slug*). Next.js posiada także integrację z Sassem — preprocesorem języka CSS.

## Sass (SCSS)



Rys. 3.9: Logo preprocesora Sass.

Sass (ang. *Syntactically Awesome Style Sheets*) to preprocesor i rozszerzenie języka CSS (ang. *Cascading Style Sheets*) posiadający dodatkowe funkcjonalności takie, jak zmienne lub możliwość zagnieżdżania stylów w sposób odwzorowujący hierarchię występującą w HTML. SCSS (ang. *Sassy CSS*) jest natomiast składnią będącą nadzbiorem CSS, co oznacza, że poprawnie napisany CSS może być jednocześnie wykorzystany w pliku SCSS [25]. Różnice pomiędzy składniami SASS

oraz SCSS zostały zaprezentowane w listingach 3.1 i 3.2. Do realizacji projektu wybrana została składnia SCSS.

```
1 div
2   p
3     font-size: 1rem
4
5   img
6     opacity: 90%
```

Listing 3.1: Przykład składni Sass.

```
1 div {
2   p {
3     font-size: 1rem;
4   }
5
6   img {
7     opacity: 90%;
8   }
9 }
```

Listing 3.2: Przykład składni SCSS.

## 4 Projekt portfolio i blogu

W tym rozdziale przedstawione zostaną założenia i wymagania (z podziałem na funkcjonalne i niefunkcjonalne) dotyczące projektu portfolio i blogu z systemem komentowania. W podrozdziale 4.4 wraz z fragmentami kodu źródłowego zaprezentowane będą również najważniejsze aspekty implementacji tworzonej strony internetowej.

### 4.1 Założenia projektu

Założenia projektu określają wizję tworzonego projektu oraz cel jego realizacji. Tworzona w ramach pracy inżynierskiej strona-wizytówka będzie składać się z dwóch części — systemu zarządzania treścią Strapi oraz odpowiedzialnej za warstwę widoku aplikacji stworzonej z wykorzystaniem szkieletu aplikacyjnego Next.js. Jej głównym zadaniem jest przedstawienie prac fotografa-dziennikarza muzycznego w formie portfolio z podziałem na projekty (np. konkretne sesje zdjęciowe lub wydarzenia) oraz udostępnienie w formie bloga przestrzeni do wypowiedzi na temat powiązany z głównym obszarem tematycznym zdjęć, którym jest muzyka na żywo. Zgodnie z wnioskami z rozdziału 2, graficzny interfejs tworzonej strony powinien być minimalistyczny, lecz nie surowy, co ma za zadanie skupić uwagę użytkownika na jej treści — postach oraz fotografiach.

### 4.2 Wymagania funkcjonalne

Wymagania funkcjonalne określają konkretne funkcje, które powinny znaleźć się w tworzonej aplikacji internetowej, aby założenia projektu zostały spełnione. Skupiają się na sposobie działania jego poszczególnych elementów, pełnią funkcję listy tego, co powinno zostać zaimplementowane [26].

## **Dostęp do systemu zarządzania treścią**

Możliwość dodawania, edytowania i usuwania elementów powinna być zarezerwowana tylko dla osoby zarządzającej stroną internetową, wyłącznie za pomocą panelu administratora. Wyjątkiem są komentarze do postów oraz wpisy kontaktowe, które mogą być dodawane za pomocą udostępnionego przez system REST API przez każdego użytkownika odwiedzającego stronę.

## **Struktura danych w systemie zarządzania treścią**

Struktura danych w systemie zarządzania treścią powinna być zorganizowana zgodnie z układem strony. Ułatwi to późniejsze pobieranie danych, a także dodawanie oraz modyfikację treści. Potrzebne będą następujące kolekcje (elementy występujące na stronie więcej, niż jeden raz):

- Post — posiada tytuł, opis, miniaturkę i tło w formie zdjęć, treść złożoną z dowolnej ilości akapitów i obrazów oraz kategorię (typ wyliczeniowy); może mieć dowolną ilość komentarzy.
- Wpis kontaktowy — zawiera dane podane przez użytkownika (imię, nazwisko, adres poczty elektronicznej) oraz wiadomość i jej temat.
- Projekt — posiada tytuł, opis, dowolną ilość zdjęć oraz miniaturkę.
- Usługa — posiada nazwę, opis, zdjęcie oraz cenę.

Na stronie pojawią się również elementy występujące wyłącznie jeden raz, dlatego wskazane będzie też stworzenie typów pojedynczych:

- Zdjęcia na stronie głównej.
- Treść strony “o mnie”, w tym nagłówek, dwa akapity oraz zdjęcie.

## **Optymalizacja strony dla wyszukiwarek internetowych (SEO)**

Aby w pełni wykorzystać możliwości tworzonej strony internetowej, powinna ona trafić do jak największej liczby odbiorców. W tym celu konieczna jest jej optymalizacja dla wyszukiwarek internetowych (ang. SEO — *Search Engine Optimization*). Można to osiągnąć poprzez wykorzystanie renderowania po stronie serwera (SSR — ang. *Server-Side Rendering*) oraz generowania statycznej strony (SSG — ang. *Static Site Generation*). Dzięki SSG oraz SSR wyszukiwarki internetowe lepiej radzą sobie z indeksowaniem stron, co zwiększa szansę na znalezienie takiej strony przez potencjalnego użytkownika. Kolejnym sposobem na optymalizację SEO jest zastosowanie unikalnego ciągu znaków w formie tzw. *sluga* do stworzenia linków do postów na blogu oraz projektów z portfolio zamiast standardowego identyfikatora występującego najczęściej po prostu w formie liczby całkowitej. Dobrą praktyką jest zawieranie jak największej liczby słów kluczowych związanych z tematyką danego postu lub przedmiotem zdjęć zamieszczonych w projekcie [27].

## **Portfolio i projekty**

Użytkownik odwiedzający projekt w portfolio fotografa powinien mieć możliwość wyświetlenia wybranego zdjęcia w większym formacie. Aby w pełni skupić jego uwagę na oglądanej fotografii, powinna ona się pojawiać w formie wyskakującego okienka z przyciemnionym lub rozmazanym tłem (którym jest strona projektu) oraz możliwością jego zamknięcia poprzez ponowne kliknięcie w dowolnym miejscu.

## **Blog i posty**

Lista postów powinna być stworzona z wykorzystaniem paginacji, aby wyeliminować problem zbyt dużej ilości miniaturek postów na stronie. Użytkownik powinien mieć możliwość wyświetlania i dodawania komentarzy do postów na blogu

w celu nawiązywania dyskusji z innymi użytkownikami i wyrażania opinii na temat treści danych postów. W przypadku braku komentarzy, pod postem powinien pojawić się napis “brak komentarzy” w celu zachęcenia osoby odwiedzającej blog do rozpoczęcia dyskusji.

## Sposoby kontaktu

Aby umożliwić podjęcie kontaktu w dowolnie wybrany przez użytkownika sposób, powinny zostać podjęte kroki ułatwiające wykonanie tej czynności. Pierwszym sposobem jest udostępnienie formularza kontaktowego, który pozwoli na wysłanie wiadomości bezpośrednio przez odwiedzaną stronę internetową. Ważną częścią formularza jest jego walidacja — użytkownik musi wiedzieć, czy wpisane przez niego dane są poprawne, a po wysłaniu — czy podjęta próba kontaktu powiodła się. Kolejne rozwiązanie to umieszczenie na stronie (np. w stopce) numeru telefonu oraz adresu poczty elektronicznej. Dane te powinny być oznaczone w strukturze strony jako kotwica (ang. *anchor*) oraz posiadać odpowiednie przedrostki (przykład dla adresu poczty elektronicznej w listingu 4.1). Ostatnim sposobem na ułatwienie kontaktu osobie odwiedzającej stronę jest podanie odnośników do mediów społecznościowych, w szczególności Instagrama oraz Facebooka, które są głównymi serwisami wykorzystywanymi w branży fotograficznej.

```
1 <div class="contacts">
2   <a href="mailto:address@example.com">address@example.com</a>
3   <a href="tel:426 313 888">42 63 13 888</a>
4 </div>
```

Listing 4.1: Przykład zastosowania znacznika *a* w języku HTML dla adresu poczty elektronicznej oraz numeru telefonu.

## 4.3 Wymagania niefunkcjonalne

W odróżnieniu od wymagań funkcjonalnych, wymagania niefunkcjonalne określają ogólne cechy tworzonej aplikacji. Dotyczą one między innymi jej jakości oraz dostępności. Skupiają się nie na tym, co robi aplikacja, lecz na tym, jak dobrze to robi [26].

### Szybkość ładowania strony, wykorzystanie SSG i SSR

Aby wywołać jak najlepsze wrażenie u użytkownika, który odwiedza tworzoną stronę internetową, powinna ona ładować się szybko, w szczególności tam, gdzie zdjęcia są jej najważniejszymi elementami (strona główna oraz portfolio). Pomocne (a zarazem przydatne do spełnienia wymagań zarówno funkcjonalnych i niefunkcjonalnych) do osiągnięcia tego celu jest wykorzystanie SSG oraz SSR, które są wbudowanymi funkcjonalnościami szkieletu aplikacyjnego Next.js.

### Wygląd strony dopasowany do jej tematyki

Tworzona strona internetowa powinna wyglądać minimalistycznie w celu skupienia uwagi odwiedzających na jej treści. Sprawdzonym rozwiązaniem, które zostało zaprezentowane w rozdziale 2, jest trzymanie się jak najmniejszej ilości kolorów. Aby dodać stronie czegoś charakterystycznego, można wykorzystać też kolor akcentowy, którym w tym przypadku jest fiolet. Przy minimalistycznym układzie i projekcie graficznym strony ważne jest również dodanie prostych animacji, np. pojawiające się podkreślenie oraz fioletowa strzałka przy najechaniu myszką na link na stronie głównej.

### Intuicyjna nawigacja i dostęp do informacji

Nawigacja jest ważną częścią projektowania stron internetowych. Powinna ona być intuicyjna i łatwa, co przy prostej strukturze strony można osiągnąć poprzez

umieszczenie widocznego cały czas paska nawigacji z linkami do wszystkich istniejących podstron. W stopce (również zawsze widocznej) powinny natomiast znaleźć się podstawowe dane kontaktowe takie jak numer telefonu oraz adres poczty elektronicznej.

## **Dostęp do strony na różnych urządzeniach i responsywność**

Ze względu na reprezentacyjną charakterystykę strony-wizytówki złożonej z portfolio oraz blogu, kluczowe jest umożliwienie jej wyświetlania na jak największej ilości urządzeń z dostępem do Internetu. Aby zapewnić wygodę przeglądania tworzonej strony, wskazane jest wykonanie jej z użyciem metody RWD (ang. *Responsive Web Design*), czyli w taki sposób, aby elementy i ich układ na stronie dostosowywały się do różnych rozmiarów ekranu [28].

## **4.4 Realizacja projektu**

W tym podrozdziale przedstawione zostały najważniejsze aspekty implementacji obu części tworzonej strony internetowej. Najpierw opisana została konfiguracja systemu zarządzania treścią Strapi, utworzenie odpowiedniej struktury danych oraz umożliwienie pobierania i dodawania treści w sposób spełniający wymagania funkcjonalne. W dalszej części rozdziału zaprezentowana jest implementacja tworzonej strony z użyciem szkieletu aplikacyjnego Next.js wraz z jej najważniejszymi funkcjonalnościami, w tym pobieraniem treści ze Strapi.

### **4.4.1 Strapi — stworzenie struktury danych oraz konfiguracja systemu zarządzania treścią**

Aby rozpocząć pracę z systemem zarządzania treścią Strapi, należy mieć zainstalowane środowisko uruchomieniowe Node.js wraz z wybranym systemem zarządzania pakietami (w tym przypadku — npm). Najprostszym sposobem na

stworzenie projektu Strapi jest skorzystanie ze skryptu instalacyjnego, a następnie przejście do katalogu głównego projektu oraz uruchomienie go. Można to zrobić, korzystając z poleceń z listingu 4.2.

```
1 npx create-strapi-app@latest portfolio-blog-strapi --quickstart
2 cd portfolio-blog-strapi
3 npm run develop
```

Listing 4.2: Polecenia tworzące oraz uruchamiające bazowy projekt systemu zarządzania treścią Strapi.

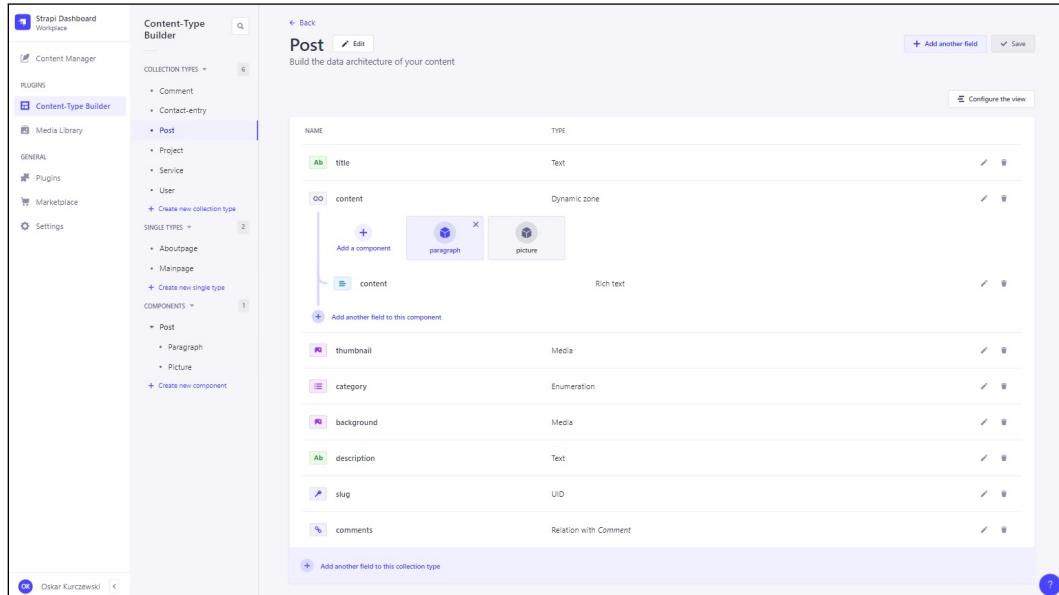
Po instalacji Strapi należy stworzyć pierwsze konto użytkownika z rolą administratora, który posiada pełne uprawnienia do systemu. Kolejnym krokiem jest utworzenie struktury danych według wymagań z rozdziału 4. Służy do tego dostępna wyłącznie w trybie rozwojowym (ang. *development mode*)<sup>4</sup>, przedstawiona na rysunku 4.1 wtyczka *Content-Type Builder* [17]. Jest to narzędzie pozwalające stworzyć strukturę danych za pomocą interfejsu użytkownika bez konieczności ręcznego pisania kodu. Przykład przedstawiony na rysunku 4.1 przedstawia interfejs narzędzia Strapi podczas używania wspomnianej wtyczki.

Stworzona struktura danych została przedstawiona w formie diagramu na rysunku 4.2. Dla ulepszenia jego czytelności została użyta następująca konwencja kolorów: elementy żółte to kolekcje, co oznacza, że mogą występować wielokrotnie (np. posty). Kolorem niebieskim zaznaczone są komponenty, które w tym przypadku składają się na treść posta w formie zaznaczonej na biało “strefy dynamicznej” (ang. *dynamic zone*) — pozwala to na stworzenie treści posta złożonego z komponentów ustawionych w dowolnej kolejności (przykładowo może to być jeden akapit, po którym następują dwa zdjęcia). Na pomarańczowo zaznaczone zostały typy pojedyncze, takie jak treść strony “o mnie” lub strony głównej. Kolor czer-

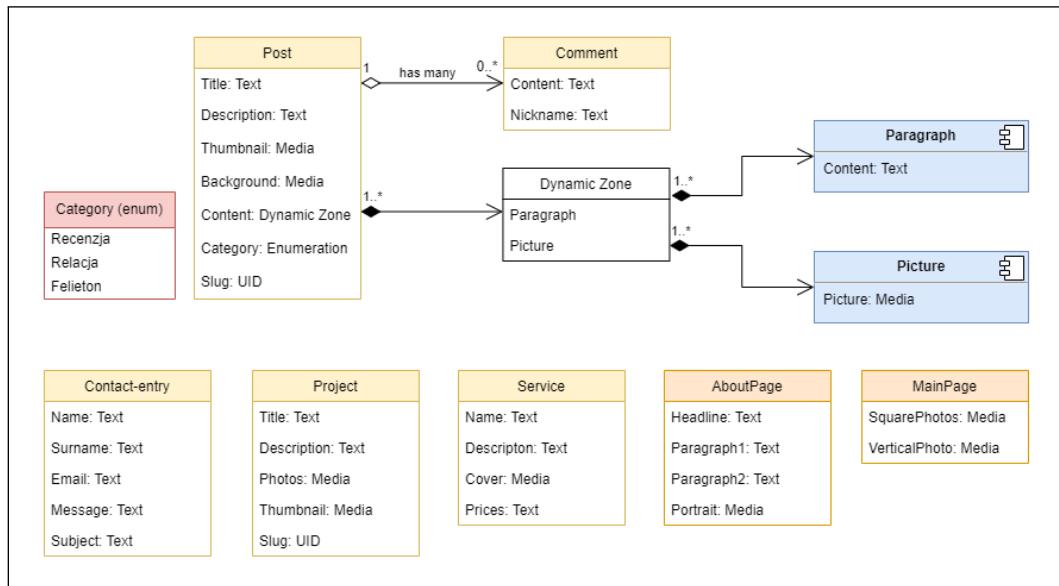
---

<sup>4</sup>Po zbudowaniu aplikacji zablokowana jest możliwość zmieniania struktury danych.

wony oznacza natomiast typ wyliczeniowy, który w tym przypadku odpowiada za kategorię, do jakiej przypisany może zostać post na blogu.



Rys. 4.1: Interfejs narzędzia Strapi z wtyczką Content-Type Builder.



Rys. 4.2: Diagram struktury danych w systemie zarządzania treścią Strapi.

Tab. 4.1: Lista uprawnień do REST API w Strapi dla użytkownika z rolą *Public* (niezalogowanego)

<b>Element</b>	<b>Uprawnienie</b>	<i>create</i>	<i>delete</i>	<i>update</i>	<i>find</i>	<i>findOne</i>
Aboutpage		n/d			X	n/d
Mainpage		n/d			X	n/d
Comment		X			X	
Contact-entry		X				
Post					X	X
Project					X	X
Service					X	X

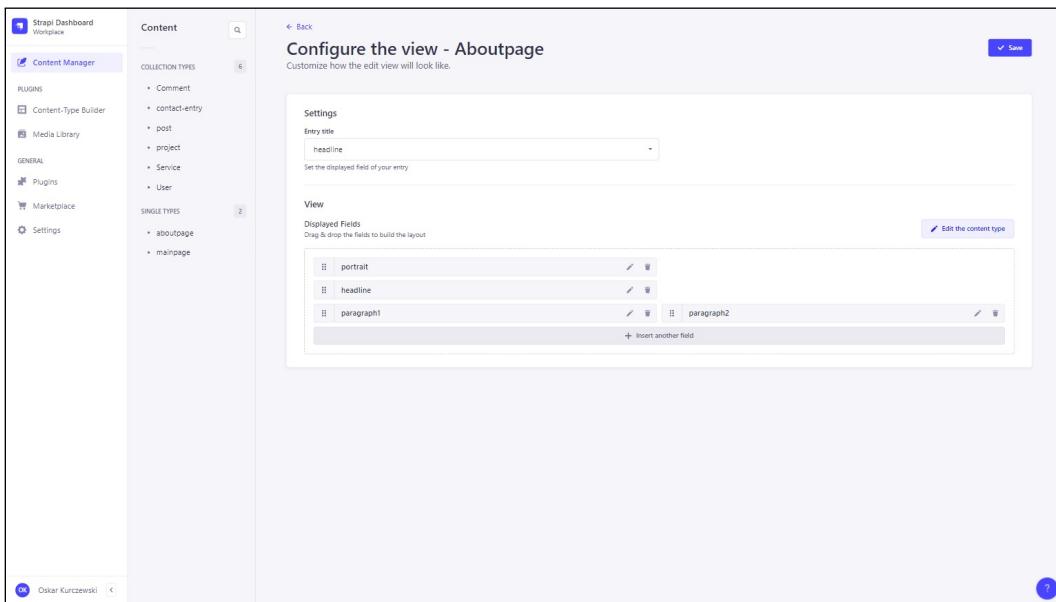
Po zakończeniu tworzenia struktury danych konieczne jest ustawienie odpowiednich uprawnień dotyczących REST API w celu umożliwienia pobierania treści do wyświetlenia na stronie oraz zablokowania możliwości ich dodawania, usuwania i modyfikacji użytkownikom nieposiadającym konta w Strapi (reprezentowanych przez rolę *Public*). Wyjątkami są komentarze, które mogą być dodawane pod postami przez każdą osobę odwiedzającą stronę oraz wpisy kontaktowe, które nie mogą być wyświetlane przez użytkowników, ale mogą być przez nich dodawane. Szczegółowa lista uprawnień dla roli *Public* zaprezentowana jest w tabeli 4.1.

Aby po stronie widoku możliwe było utworzenie linków do postów oraz projektów w postaci unikalnego ciągu znaków, konieczne było wprowadzenie modyfikacji do kontrolera dla wspomnianych kolekcji, a dokładniej — nadpisanie funkcji pobrania jednego elementu (*findOne*). Aby to zrobić, należy podać klucz, według którego ma być szukany dany element: jest nim właśnie *slug*. Aby dane były pobierane w odpowiedni sposób trzeba ustawić parametr *populate* dla zawartości elementu na wartość *true*. Przykład modyfikacji kontrolera znajduje się w listingu 4.3.

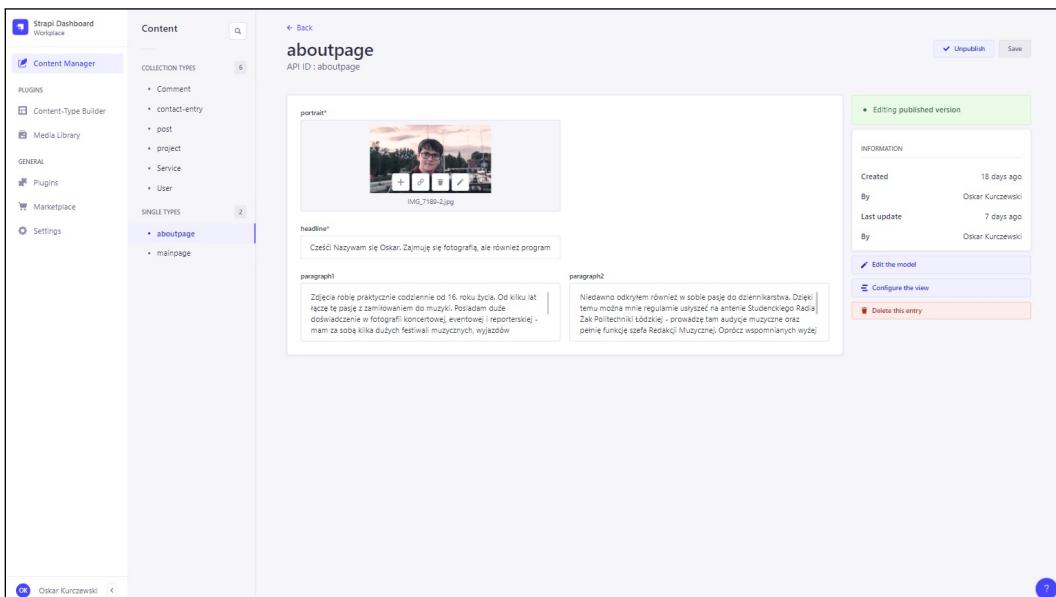
Ostatnim (choć wpływającym jedynie na wygodę dodawania treści) krokiem jest skorzystanie z funkcjonalności Strapi pozwalającej na rozmieszczenie elementów w edytorze treści w taki sposób, który odpowiada układowi faktycznej strony (rys. 4.3). Dzięki temu można rozpoczęć dodawanie treści w prosty i intuicyjny sposób (rys. 4.4).

```
1 module.exports = createCoreController("api::post.post", ({ strapi }) => ({
2   async findOne(ctx) {
3     const { id } = ctx.params;
4
5     const entity = await strapi.db.query("api::post.post").findOne({
6       where: { slug: id },
7       populate: {
8         content: {
9           populate: {
10             picture: true,
11             },
12             },
13             thumbnail: true,
14             background: true,
15             comments: true,
16             },
17           });
18
19     const sanitizedEntity = await this.sanitizeOutput(entity, ctx);
20
21     return this.transformResponse(sanitizedEntity);
22   },
23 })
```

Listing 4.3: Przykład modyfikacji do kontrolera REST API w systemie zarządzania treścią Strapi.



Rys. 4.3: Rozmieszczanie elementów edytora treści w systemie zarządzania treścią Strapi.



Rys. 4.4: Efekt rozmieszczenia elementów w edytorze treści w systemie zarządzania treścią Strapi.

#### 4.4.2 Next.js — implementacja strony internetowej

Aby rozpocząć implementację aplikacji z wykorzystaniem Next.js, należy stworzyć i uruchomić nowy projekt z użyciem polecenia z listingu 4.4. Konieczna jest też instalacja preprocesora Sass za pomocą menadżera pakietów npm. Po jej zakończeniu można rozpoczęć pracę nad implementacją strony internetowej.

```
1 npx create-next-app@latest portfolio-blog-next
2 cd portfolio-blog-next
3 npm run dev
```

Listing 4.4: Polecenie tworzące oraz uruchamiające nowy projekt z użyciem szkieletu aplikacyjnego Next.js.

### Utworzenie projektu i organizacja jego struktury

Pierwszy krok to stworzenie plików odpowiadających odpowiednim podstrogramom w katalogu `/pages` — szkielet aplikacyjny Next.js korzysta z routingu opartego na stronach będących komponentami biblioteki React [24]. Potrzebne będą strona główna, strony portfolio i blogu (listy dodanych projektów/postów oraz widoki poszczególnych projektów/postów), strona z listą usług świadczonych przez fotografa, strona “o mnie” oraz strona kontaktowa. Warto również dodać własną stronę “nie znaleziono”, która jest wyświetlana, gdy dany link nie istnieje. Listing 4.5 przedstawia drzewo plików w katalogu `/pages`. Warto w nim zwrócić uwagę na nawiasy kwadratowe przy plikach `blog/[slug].js` oraz `portfolio/[slug].js` — oznacza to, że są to ścieżki dynamiczne, tzn. linki do postów oraz projektów generują się na podstawie parametru w postaci `sluga` pobranego z systemu zarządzania treścią Strapi.

```
1 pages
2 | 404.js
3 | about.js
4 | contact.js
5 | index.js
6 | services.js
7 | _app.js
8 | _document.js
9 |
10|---blog
11|   index.js
12|   [slug].js
13|
14|---portfolio
15|   index.js
16|   [slug].js
```

Listing 4.5: Drzewo plików w katalogu `/pages`.

### Pobieranie danych z systemu zarządzania treścią

Po utworzeniu odpowiedniej struktury plików w projekcie, można przejść do budowy układu poszczególnych stron oraz pobierania danych z systemu zarządzania treścią Strapi. Jako przykład posłuży strona “o mnie” znajdująca się w pliku `about.js`. Widoczny w listingu 4.6 komponent funkcyjny `About` zwraca zawartość strony “o mnie” w postaci kodu JSX. Funkcja `getStaticProps` przedstawiona w listingu 4.7 służy natomiast do pobierania danych z systemu zarządzania treścią i statycznego generowania strony (SSG), dzięki czemu załadowuje się ona szybciej oraz będzie lepiej indeksowana w wyszukiwarkach internetowych. W przypadku danych, które po stronie systemu zarządzania treścią mogą być modyfikowane lub dodawane często, powinno się używać funkcji `getServerSideProps`, oferującej renderowanie po stronie serwera (SSR). Różnica pomiędzy `getStaticProps` a `getServerSideProps` jest taka, że przy używaniu `getStaticProps` zawartość strony zostanie wygenerowana w trakcie jej budowania, gdzie w przypadku `getServerSideProps`

renderowanie po stronie serwera ma miejsce za każdym razem, kiedy użytkownik wykona żądanie [24]. Dzięki temu nie jest konieczne ponowne uruchomienie aplikacji Next.js w celu wyświetlenia nowo dodanych treści, co jest wymagane jeśli korzysta się z SSG.

```
1  export default function About({data}) {
2    const url = "http://localhost:1337";
3    return (
4      <>
5        <Head>
6          <title>Oskar Kurczewski - o mnie</title>
7        </Head>
8        <div className='about-container'>
9          <div className='image-container'>
10            <Image src={url + data.data.attributes.portrait.data.attributes.url}
11                  fill alt={data.data.attributes.portrait.data.attributes.name}/>
12          </div>
13          <div>
14            <div className='text'>
15              <h1>{data.data.attributes.headline}</h1>
16              <p>
17                {data.data.attributes.paragraph1}
18              </p>
19              <p>
20                {data.data.attributes.paragraph2}
21              </p>
22            </div>
23          </div>
24        </>
25      )
26    }
```

Listing 4.6: Fragment pliku `about.js` przedstawiający komponent funkcyjny `About`.

REST API systemu zarządzania treścią Strapi zwraca dane w formacie JSON (*JavaScript Object Notation*). Listing 4.8 przedstawia przykładowe, trywialne dane do strony “o mnie”. Można je przekazać do komponentu *About* w formie właściwości (ang. *props*), a następnie wyświetlić na stronie poprzez wskazanie konkretnego pola, czego przykładem jest linijka czternasta w listingu 4.6 odpowiadająca za wyświetlanie nagłówka.

```
1 export async function getStaticProps() {
2   const res = await fetch(`#${process.env.API_URL}/api/aboutpage?populate=*`)
3   const data = await res.json()
4   return { props: { data } }
5 }
```

Listing 4.7: Fragment pliku `about.js` przedstawiający funkcję `getStaticProps`.

```
1 {
2   "data": {
3     "id": 1,
4     "attributes": {
5       "headline": "Headline example",
6       "paragraph1": "Paragraph 1 example",
7       "paragraph2": "Paragraph 2 example",
8       "createdAt": "2023-01-15T20:23:44.504Z",
9       "updatedAt": "2023-01-26T22:01:07.272Z",
10      "publishedAt": "2023-01-15T20:23:45.638Z"
11    }
12  },
13  "meta": {}
14}
```

Listing 4.8: Dane w formacie JSON zwrócone przez Strapi dla strony “o mnie”.

## Komponenty

Aby zgodnie z zasadą pojedynczej odpowiedzialności [29] polepszyć czytelność kodu oraz ułatwić jego późniejszą ewentualną modyfikację, zostały stworzone następujące komponenty:

- *Card* — znajdująca się na liście postów karta będąca jednocześnie odnośnikiem do strony z postem. Zawiera miniaturkę, kategorię, tytuł i datę publikacji.
- *Comment* — komponent reprezentujący komentarz pod postem. Składa się z nazwy użytkownika, treści komentarza oraz czasu, który upłynął od jego publikacji (do jego obliczenia wykorzystana została prosta biblioteka *react-timeago* zainstalowana z użyciem menadżera pakietów npm).
- *ImageModal* — okno modalne zajmujące cały ekran, służące do wyświetlania zdjęcia z projektu w portfolio w dużym formacie.
- *Service* — komponent w formie karty zawierającej informacje na temat danej usługi (opis i cena), widoczne po najechaniu na obrazek, pod którym widnieje nazwa usługi.
- *ProjectThumbnail* — miniaturka projektu składająca się ze zdjęcia, opisu oraz przycisku “więcej zdjęć” przenoszącego do widoku całego projektu.
- *Footer* — stopka strony składająca się z informacji kontaktowych (telefon, adres poczty elektronicznej) oraz odnośników do profilów fotografa w portałach społecznościowych.
- *Navbar* — pasek z nagłówkiem strony w postaci imienia i nazwiska fotografa oraz główne dziedziny świadczonych usług fotograficznych (w przy-

padku strony głównej) lub pasek nawigacyjny jeśli użytkownik znajduje się w innym miejscu.

## Komentarze i formularz kontaktowy — przesyłanie danych do Strapi

Aby umożliwić dodawanie komentarzy pod postami oraz wysyłanie wiadomości poprzez formularz kontaktowy, należy w odpowiedni sposób przesłać dane do systemu zarządzania treścią Strapi z użyciem metody POST protokołu HTTP. Można to zrobić, używając funkcji *fetch* z odpowiednimi parametrami tak, jak w listingu 4.9. Owe parametry to *method*, *headers* oraz *body*. *Method* określa metodę HTTP (w tym przypadku POST), *headers* to nagłówki HTTP do wysłania z żądaniem (tutaj: określające sposób przekazania danych), natomiast *body* to przesyłane dane (w formacie JSON). Przy wysyłaniu żądania ważne jest również określenie jego adresu, którym w tym przypadku jest `/api/comments`.

```
1 const commentData = {
2   nickname: nickname,
3   content: content,
4   post: {
5     id: postId
6   }
7 }
8
9 const add = await fetch(`#${url}/api/comments`, {
10   method: "POST",
11   headers: {
12     'Accept': 'application/json',
13     'Content-Type': 'application/json'
14   },
15   body: JSON.stringify({data: commentData})
16 })
```

Listing 4.9: Fragment funkcji *addComment* dodającej komentarz do postu.

## Walidacja wprowadzanych danych

Ze względu na to, że dane są przesyłane za pomocą formularza, należy sprawdzić poprawność wprowadzonych przez użytkownika danych za pomocą walidacji. W przypadku komentarza wystarczy atrybut *required*, który blokuje możliwość wysłania danych jeśli pola treści oraz ksywki nie są wypełnione. Przy formularzu kontaktowym oprócz obowiązkowych pól, należy dodatkowo sprawdzić prawidłowość podawanego adresu poczty elektronicznej. Zostało to zrobione na dwa sposoby: w samym formularzu z wykorzystaniem atrybutu “`type="email"`” oraz w systemie zarządzania treścią Strapi poprzez zastosowanie wyrażenia regularnego.

## Responsive Web Design

Jednym z wymagań niefunkcjonalnych projektu jest zastosowanie metody RWD (ang. *Responsive Web Design*) w celu dostosowania tworzonej strony internetowej do ekranów różnych rozmiarów. Preprocesor Sass pozwala na zrobienie tego w bardzo prosty sposób, którym jest wykorzystanie zmiennych oraz domieszek (*mixin*) — fragmentów kodu SCSS pozwalających na używanie ich wielokrotnie bez konieczności ich kopiowania. Listing 4.10 przedstawia deklarację zmiennej określającej szerokość ekranu średniej wielkości (np. tabletów) oraz domieszki pozwalające ustawać dla nich indywidualne zasady wyświetlania w sposób ukazany w listingu 4.11.

```
1 $screen-md-min: 768px;
2 @mixin md {
3   @media (min-width: #{$screen-md-min}) {
4     @content;
5   }
6 }
```

Listing 4.10: Przykład deklaracji domieszki w SCSS.

```
1 @include sm {
2     width: 350px;
3     margin-top: 20px;
4     margin-bottom: 0px;
5 }
6 @include md {
7     width: 600px;
8     margin-top: 30px;
9     margin-bottom: 0px;
10 }
11 @include lg {
12     width: 1100px;
13     margin-top: 50px;
14     margin-bottom: 120px;
15 }
```

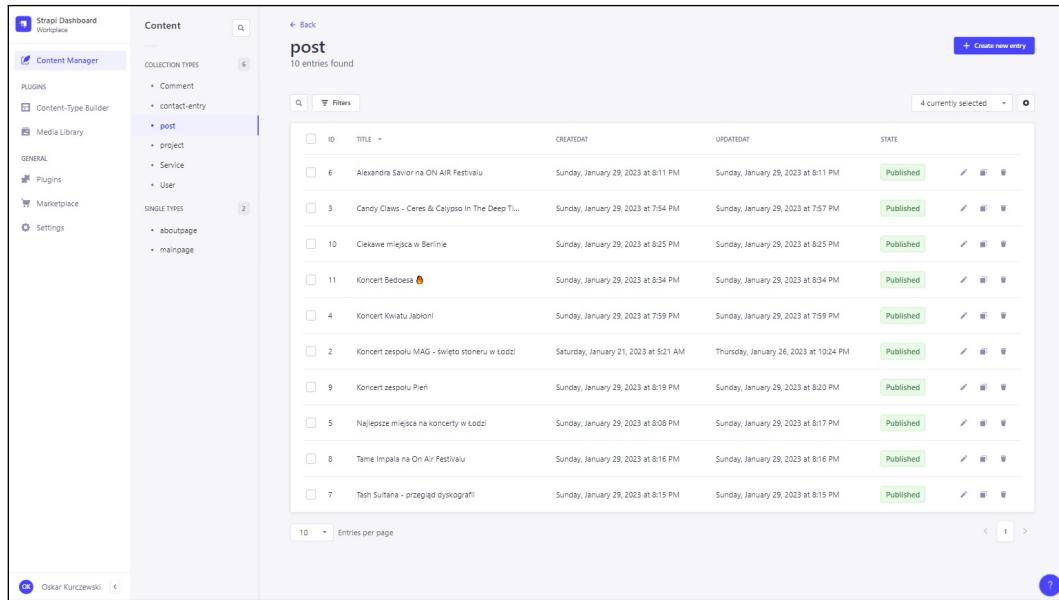
Listing 4.11: Przykład użycia domieszki w SCSS.

## 5 Portfolio i blog

W tym rozdziale zaprezentowany zostanie efekt końcowy oraz sposób użytkowania stworzonej strony internetowej z perspektywy administratora (podrozdział 5.1) oraz użytkownika odwiedzającego stronę (podrozdział 5.2). Opisane będzie między innymi korzystanie z systemu zarządzania treścią (tworzenie, edycja i usuwanie zawartości strony, wyświetlanie wiadomości z formularza kontaktowego), dodawanie komentarzy i dostosowanie strony do ekranów różnej wielkości.

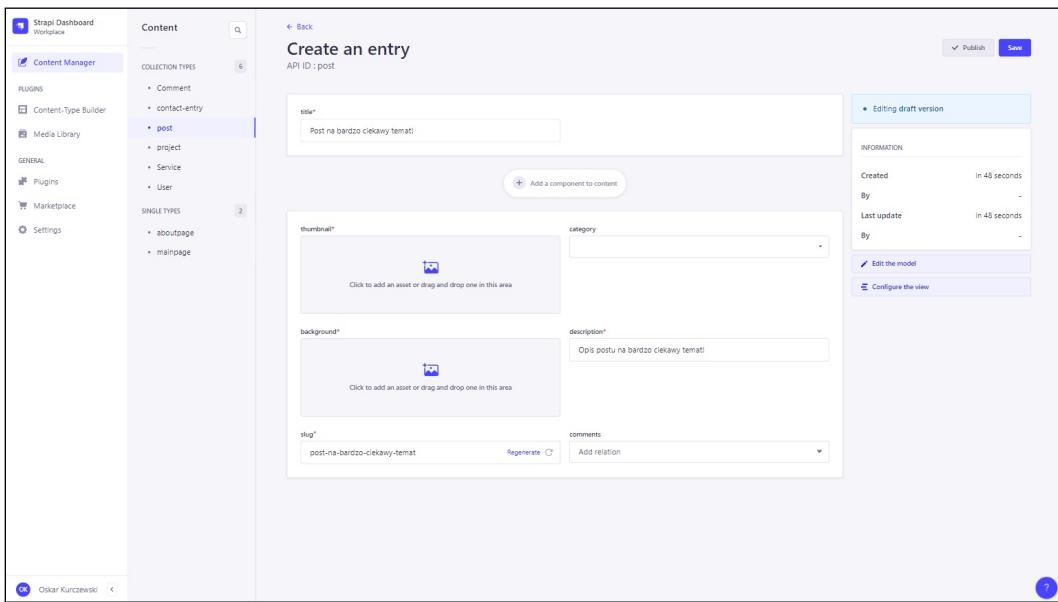
### 5.1 System zarządzania treścią

Jednym z dwóch głównych elementów tworzonej strony internetowej jest system zarządzania treścią, którym jest Strapi. Aby z niego skorzystać, należy zalogować się do panelu administracyjnego, gdzie dostępny jest menedżer treści (ang. *Content Manager*) pozwalający na dodawanie, edycję i usuwanie treści, które są widoczne dla osób odwiedzających stronę.

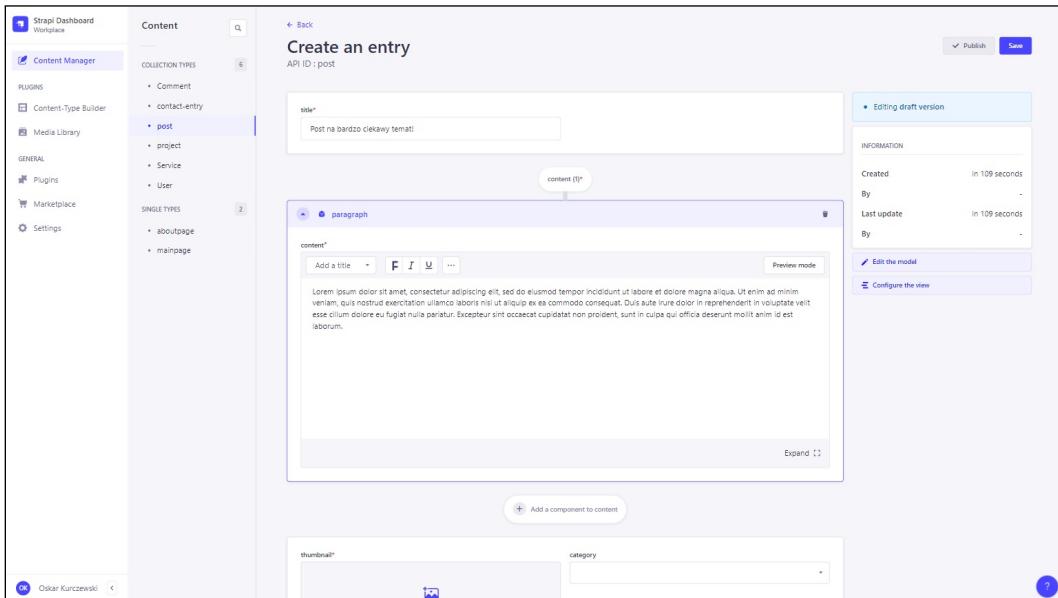


The screenshot shows the Strapi Content Manager interface. On the left, there's a sidebar with navigation links: Strapi Dashboard, Content Manager (which is selected and highlighted in blue), Content-Type Builder, Media Library, Plugins, Marketplace, and Settings. Under Content Manager, there are sections for 'COLLECTION TYPES' (Comment, contact-entry, post, project, Service, User) and 'SINGLE TYPES' (aboutpage, mainpage). The main area is titled 'post' and shows a list of 10 entries found. The columns in the table are ID, TITLE, CREATEDAT, UPDATEDAT, and STATE. Each entry has a checkbox next to it. The first entry is 'Alexandra Savori na ON AIR Festiwalu'. The last entry is 'Tash Sultana - przegląd dyskografii'. All entries have a 'Published' status. There are also icons for edit, delete, and preview. At the bottom, there's a dropdown for 'Entries per page' set to 10, and a footer with the name 'Oskar Kurczewski'.

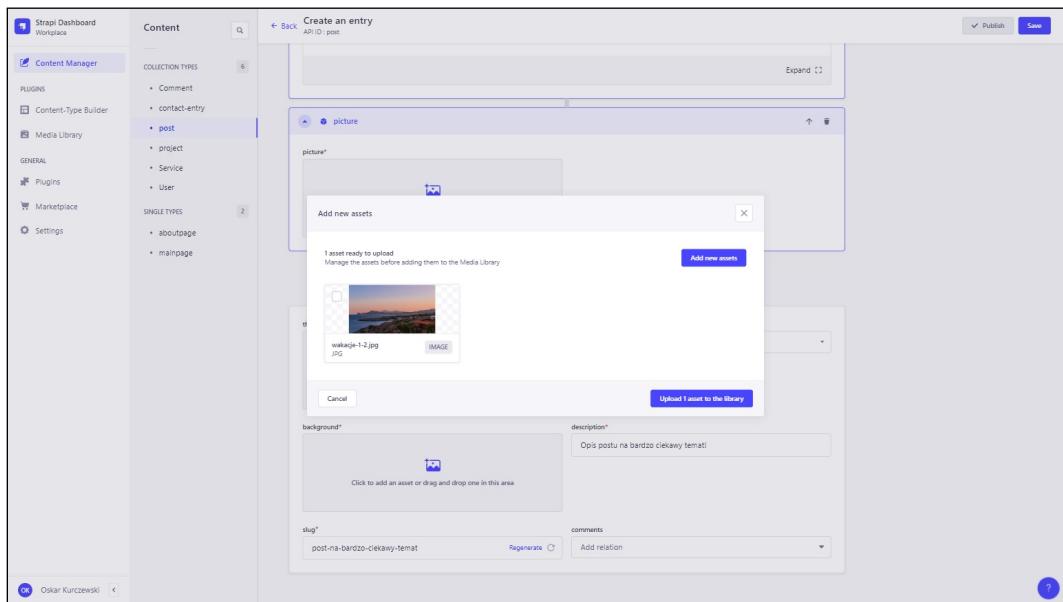
Rys. 5.1: Widok menedżera treści w Strapi (lista postów).



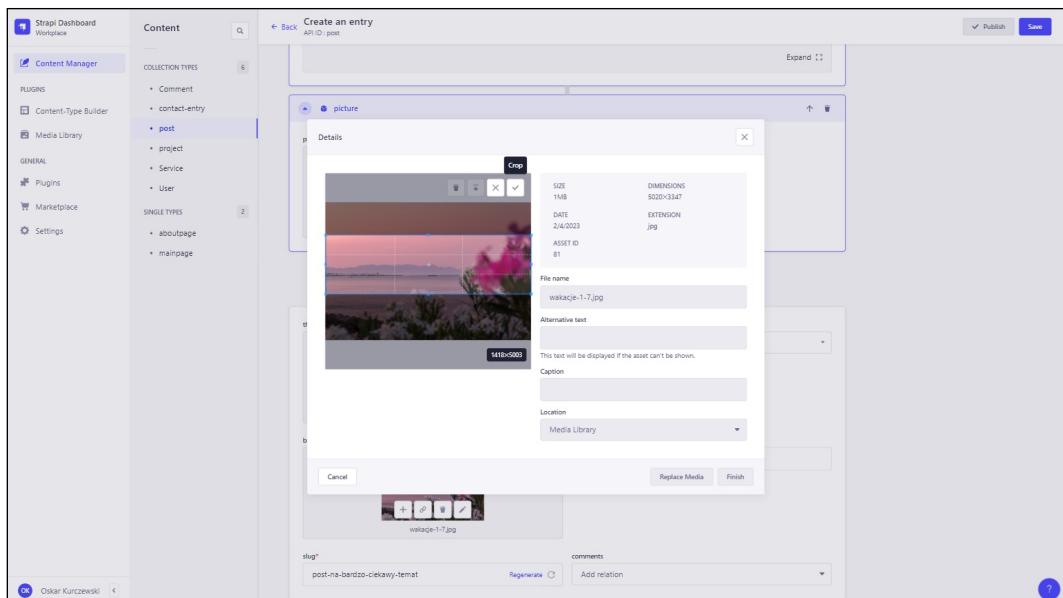
Rys. 5.2: Widok menadżera treści w Strapi (lista postów).



Rys. 5.3: Widok menadżera treści w Strapi (tworzenie nowego postu).



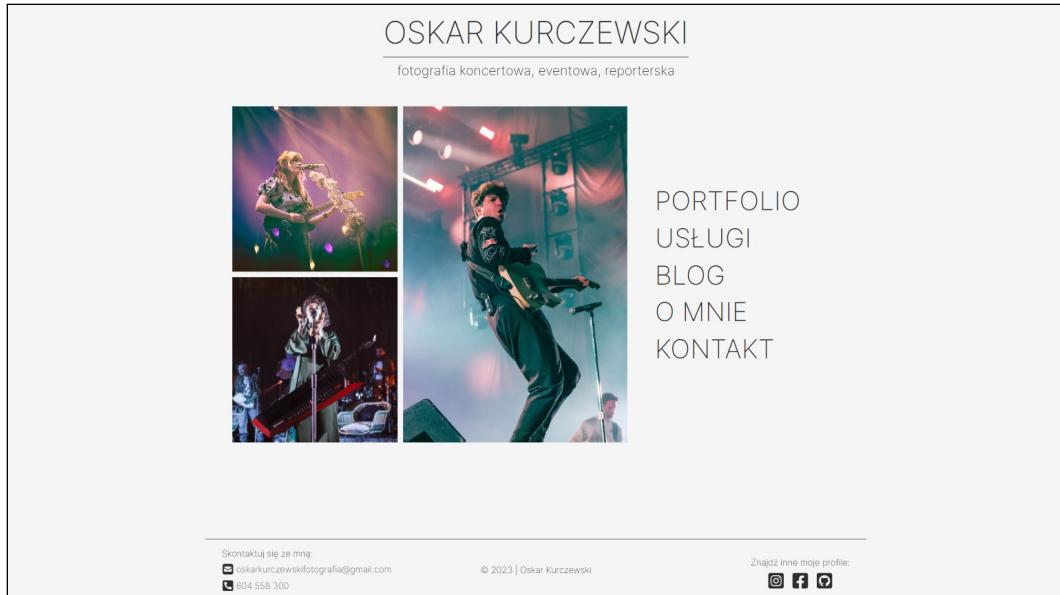
Rys. 5.4: Widok menadżera treści w Strapi (dodawanie nowego obrazu).



Rys. 5.5: Widok menadżera treści w Strapi (przykład edycji obrazu).

## 5.2 Strona internetowa

Strona internetowa z portfolio oraz blogiem jest drugą, widoczną dla dowolnego użytkownika częścią tworzonego projektu.



Rys. 5.6: Strona główna.

**OSKAR KURCZEWSKI**

PORTFOLIO • USŁUGI • BLOG • O MNIE • KONTAKT



Kwiat Jabłoni - 27.02.2022  
Klub Wytwórnia, Łódź

więcej zdjęć



Skontaktuj się ze mną:  
oskar.kurczewski.fotografia@gmail.com  
604 558 300

© 2023 | Oskar Kurczewski

Znajdź inne moje profile:  
[Instagram](#) [Facebook](#) [Twitter](#)

Rys. 5.7: Portfolio — widok listy projektów.

**OSKAR KURCZEWSKI**

PORTFOLIO • USŁUGI • BLOG • O MNIE • KONTAKT

Koncert zespołu MAG - 14.05.2022  
Muzeum Książki Artystycznej, Łódź

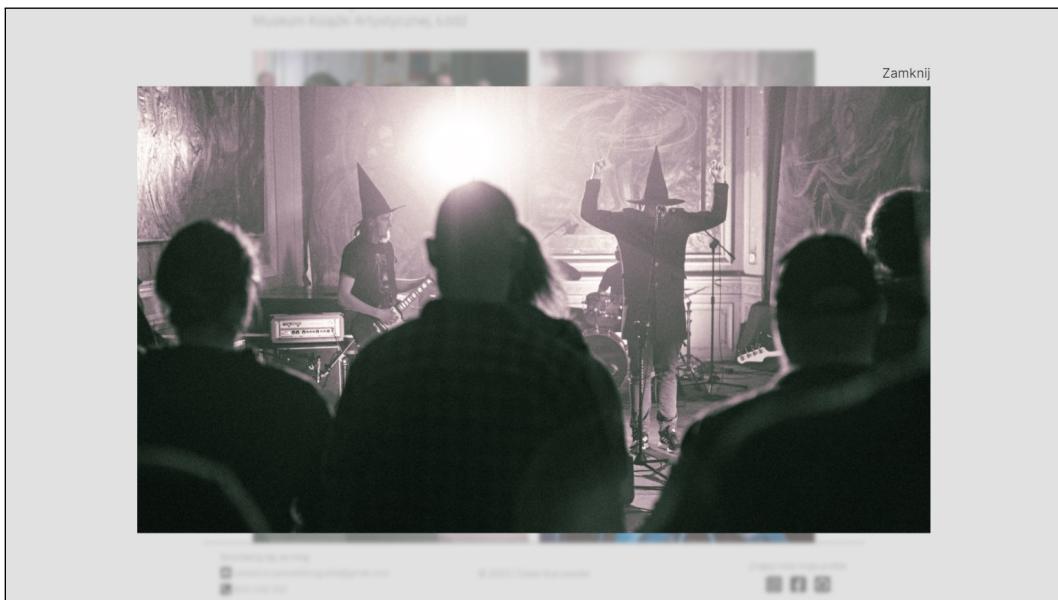


Skontaktuj się ze mną:  
oskar.kurczewski.fotografia@gmail.com  
604 558 300

© 2023 | Oskar Kurczewski

Znajdź inne moje profile:  
[Instagram](#) [Facebook](#) [Twitter](#)

Rys. 5.8: Portfolio — widok projektu.



Rys. 5.9: Portfolio — widok zdjęcia w większym formacie.

The screenshot shows a website for 'OSKAR KURCZEWSKI'. The header includes the name and navigation links: PORTFOLIO • USŁUGI • BLOG • O MNIE • KONTAKT. Below the header are three image thumbnails: 'Muzyka na żywo' (Live music), 'Sesje zdjęciowe' (Photo sessions), and 'Imprezy okolicznościowe' (Event photography). To the right of the thumbnails is a text box with promotional text and a price. At the bottom, there are contact details and social media links.

OSKAR KURCZEWSKI

PORTFOLIO • USŁUGI • BLOG • O MNIE • KONTAKT

Muzyka na żywo      Sesje zdjęciowe      Imprezy okolicznościowe

Urodziny, potańcówki, rocznice, śluby, czy też rozwody - jeśli jest okazja, żeby coś świętować, to jest okazja, żebym zrobił tam zdjęcia!

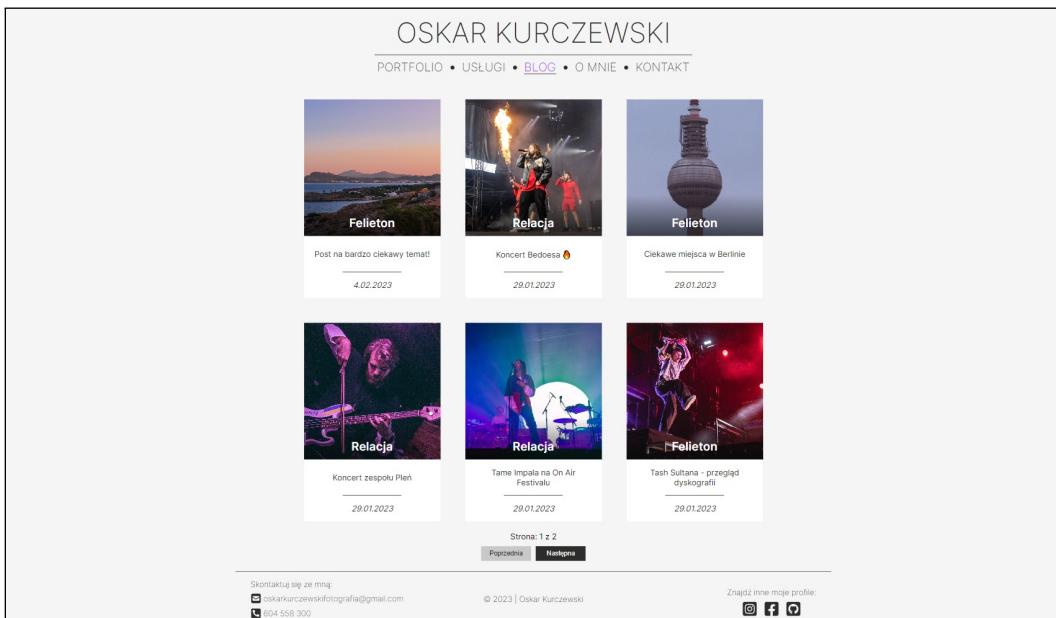
Ceny:  
od 600zł

Skontaktuj się ze mną  
oskar.kurczewski.fotografia@gmail.com  
804 558 300

© 2023 | Oskar Kurczewski

Znajdź inne moje profile:  
[Instagram](#) [Facebook](#) [Twitter](#)

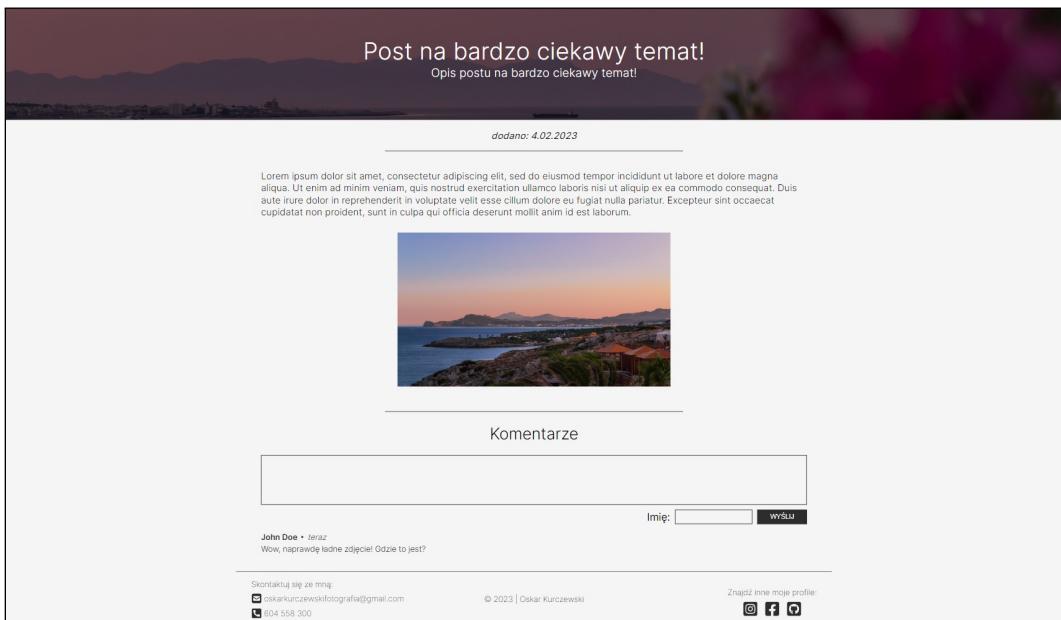
Rys. 5.10: Strona z listą usług po najechaniu na kartę z usługa "imprezy okolicznościowe".



Rys. 5.11: Blog — lista postów.



Rys. 5.12: Blog — widok postu.



Rys. 5.13: Blog — widok postu z komentarzem.



Rys. 5.14: Strona “o mnie”.

**OSKAR KURCZEWSKI**

PORTFOLIO • USŁUGI • BLOG • O MNIE • [KONTAKT](#)

Skontaktuj się za pomocą formularza:

Imię: \*  Nazwisko: \*   
Email: \*   
Temat: \*   
Wiadomość: \*

Pomyślnie wysłano wiadomość

...lub skorzystaj z innych sposobów:

---

Skontaktuj się ze mną:  
✉ oskar.kurczewski.fotografia@gmail.com  
📞 604 558 300

© 2023 | Oskar Kurczewski

Znajdź inne moje profile:

Rys. 5.15: Formularz kontaktowy.

## **6 Podsumowanie**

# Spis rysunków

2.1	<i>www.ashleyosborn.com</i> — strona główna.	8
2.2	<i>www.ashleyosborn.com</i> — sekcja “o mnie”.	9
2.3	<i>www.ashleyosborn.com</i> — formularz kontaktowy.	9
2.4	<i>www.ashleyosborn.com</i> — blog (lista postów).	11
2.5	<i>www.ashleyosborn.com</i> — blog (widok postu).	11
2.6	<i>www.zackerymichaelstudio.com</i> — strona główna.	12
2.7	<i>www.zackerymichaelstudio.com</i> — sekcja kontaktowa.	12
2.8	Instagram — profil użytkownika.	15
2.9	Instagram — widok postu.	15
2.10	Facebook — widok profilu użytkownika ze zdjęciem w tle.	17
2.11	Facebook — widok tablicy użytkownika po przewinięciu.	17
2.12	Facebook — widok postu ze zdjęciem.	18
2.13	Facebook — widok postu tekstowego (z osadzonym linkiem).	18
3.1	Logo narzędzia Figma.	20
3.2	Interfejs użytkownika narzędzia Figma.	20
3.3	Logo języka JavaScript.	21
3.4	Logotypy Node.js oraz npm.	21
3.5	Logo systemu zarządzania treścią Strapi.	22
3.6	Logo systemu zarządzania bazami danych SQLite.	23
3.7	Logo biblioteki React.	23
3.8	Logo szkieletu aplikacyjnego Next.js.	24
3.9	Logo preprocesora Sass.	24
4.1	Interfejs narzędzia Strapi z wtyczką Content-Type Builder.	33
4.2	Diagram struktury danych w systemie zarządzania treści Strapi.	33
4.3	Rozmieszczanie elementów edytora treści w systemie zarządzania treścią Strapi.	36

4.4	Efekt rozmieszczenia elementów w edytorze treści w systemie zarządzania treścią Strapi. . . . .	36
5.1	Widok menadżera treści w Strapi (lista postów). . . . .	45
5.2	Widok menadżera treści w Strapi (lista postów). . . . .	46
5.3	Widok menadżera treści w Strapi (tworzenie nowego postu). . . . .	46
5.4	Widok menadżera treści w Strapi (dodawanie nowego obrazu). . . . .	47
5.5	Widok menadżera treści w Strapi (przykład edycji obrazu). . . . .	47

## **Spis tabel**

4.1 Lista uprawnień do REST API w Strapi dla użytkownika z rolą <i>Public</i> (niezalogowanego) . . . . .	34
--------------------------------------------------------------------------------------------------------------	----

# Spis listingów

3.1	Przykład składni Sass. . . . .	25
3.2	Przykład składni SCSS. . . . .	25
4.1	Przykład zastosowania znacznika <i>a</i> w języku HTML dla adresu poczty elektronicznej oraz numeru telefonu. . . . .	29
4.2	Polecenia tworzące oraz uruchamiające bazowy projekt systemu zarządzania treścią Strapi. . . . .	32
4.3	Przykład modyfikacji do kontrolera REST API w systemie zarządzania treścią Strapi. . . . .	35
4.4	Polecenie tworzące oraz uruchamiające nowy projekt z użyciem szkieletu aplikacyjnego Next.js. . . . .	37
4.5	Drzewo plików w katalogu <i>/pages</i> . . . . .	38
4.6	Fragment pliku <i>about.js</i> przedstawiający komponent funkcyjny <i>About</i> . . . . .	39
4.7	Fragment pliku <i>about.js</i> przedstawiający funkcję <i>getStaticProps</i> . .	40
4.8	Dane w formacie JSON zwrócone przez Strapi dla strony “o mnie”.	40
4.9	Fragment funkcji <i>addComment</i> dodającej komentarz do postu. . . .	42
4.10	Przykład deklaracji domieszki w SCSS. . . . .	43
4.11	Przykład użycia domieszki w SCSS. . . . .	44

## Bibliografia

- [1] Leszek J. Pękalski, *Kalejdoskop fotografii. Między techniką a sztuką*, Helion, 2012.
- [2] The Nobel Prize in Physics 2009, <https://www.nobelprize.org/prizes/physics/2009/summary/>, dostęp 27.01.2023.
- [3] The first digital photos, National Science and Media Museum, <https://www.scienceandmediamuseum.org.uk/objects-and-stories/first-digital-photos>, dostęp 27.01.2023.
- [4] Most followers on Instagram, Statista, <https://www.statista.com/statistics/421169/most-followers-instagram/>, dostęp 27.01.2023.
- [5] Global social networks ranked by number of users, Statista, <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>, dostęp 27.01.2023
- [6] Ashley Osborn, <https://www.ashleyosborn.com/>, dostęp 27.01.2023.
- [7] State of the Photography Industry Report 2022, Format, <https://www.format.com/magazine/features/photography/state-of-the-photography-industry-2022>, dostęp 27.01.2023.
- [8] Jason Beaird, James George, *Niezawodne zasady web designu. Projektowanie spektakularnych witryn internetowych. Wydanie III*, Helion, 2015.
- [9] Jennifer Niederst Robbins, *Projektowanie stron internetowych. Przewodnik dla początkujących webmasterów po HTML5, CSS3 i grafice. Wydanie IV*, Helion, 2014.

- [10] Line Length Readability, Baymard, <https://baymard.com/blog/line-length-readability>, dostęp 27.01.2023
- [11] Marcin Żukowski, *Twoja firma w social mediach. Podręcznik marketingu internetowego dla małych i średnich przedsiębiorstw*, Helion, 2016.
- [12] Fabio Staiano, *Designing and Prototyping Interfaces with Figma*, Packt Publishing, 2022.
- [13] David Flanagan, *JavaScript. Przewodnik. Poznaj język mistrzów programowania. Wydanie VII*, Helion, 2021.
- [14] Facebook Overview, History & Facts, Britannica, <https://www.britannica.com/topic/Facebook>, dostęp 27.01.2023.
- [15] Introducing Instagram Stories, <https://about.instagram.com/blog/announcements/introducing-instagram-stories>, dostęp 27.01.2023.
- [16] A Brief History of Instagram's Fateful First Day, Time, <https://time.com/4408374/instagram-anniversary/>, dostęp 27.01.2023.
- [17] Strapi Developer Docs, <https://docs.strapi.io/developer-docs/latest/getting-started/introduction.html>, dostęp 27.01.2023.
- [18] Strapi Features, <https://strapi.io/features>, dostęp 27.01.2023.
- [19] About Node.js, <https://nodejs.org/en/about/>, dostęp 27.01.2023.
- [20] About npm, <https://www.npmjs.com/about>, dostęp 27.01.2023.
- [21] About SQLite, <https://www.sqlite.org/about.html>, dostęp 27.01.2023.
- [22] React - A JavaScript library for building user interfaces, <https://reactjs.org>, dostęp 27.01.2023.

- [23] Next.js by Vercel - the React framework, <https://nextjs.org>, dostęp 27.01.2023.
- [24] Next.js Documentation, <https://nextjs.org/docs/getting-started>, dostęp 27.01.2023.
- [25] Sass basics, <https://sass-lang.com/guide>, dostęp 27.01.2023.
- [26] Karl E Wiegers, Joy Beatty, *Specyfikacja oprogramowania. Inżynieria wymagań. Wydanie III*, Helion, 2014.
- [27] Eric Enge, Stephan Spencer, Jessie Stricchiola, *SEO, czyli sztuka optymalizacji witryn dla wyszukiwarek*, Helion, 2016.
- [28] Thoriq Firdaus, *Responsive Web Design. Nowoczesne strony WWW na przykładach*, Helion, 2014.
- [29] Gary McLean Hall, *Adaptywny kod. Zwinne programowanie, wzorce projektowe i SOLID-ne zasady. Wydanie II*, Helion, 2017.