

Plan Projektu: Strona Wizytówki / Portfolio

1. Opis Projektu

Cel:

Celem projektu jest stworzenie nowoczesnej, responsywnej strony wizytówki/portfolio prezentującej dotychczasowe projekty, umiejętności oraz dane kontaktowe. Strona ma budować profesjonalny wizerunek oraz umożliwiać łatwą aktualizację treści.

Zakres:

- Strona główna z krótkim bio i sekcją "O mnie"
- Portfolio z prezentacją projektów
- Sekcja CV lub szczegółowych umiejętności
- Formularz kontaktowy
- Integracja z mediami społecznościowymi
- Opcjonalnie: Blog lub sekcja aktualności

2. Funkcjonalności i Wymagania

2.1. Strona Główna

- Opis funkcji:
Widok główny, który zawiera przegląd najważniejszych informacji: zdjęcie profilowe, krótkie bio, linki do mediów społecznościowych oraz przyciski nawigacyjne do poszczególnych sekcji.
- Przykład użycia:
Użytkownik otwiera stronę i szybko uzyskuje informacje o autorze, przegląda portfolio oraz przechodzi do sekcji "Kontakt".

2.2. Portfolio / Projekty

- Opis funkcji:
Lista lub siatka projektów z miniaturami, opisami, linkami do szczegółów (np. GitHub, live demo) oraz ewentualną galerią zdjęć lub wideo.
- Przykład użycia:
Użytkownik klika na projekt, aby zobaczyć szczegółowe informacje dotyczące realizacji, użytych technologii i wyniku.

2.3. Sekcja "O mnie" / CV

- Opis funkcji:
Rozbudowany opis doświadczenia, umiejętności, certyfikatów oraz historii kariery. Może zawierać plik do pobrania (PDF).
- Przykład użycia:
Użytkownik zapoznaje się z doświadczeniem autora, aby ocenić jego kompetencje.

2.4. Formularz kontaktowy

- **Opis funkcji:**
Prostota kontaktu poprzez formularz umożliwiający wysłanie wiadomości e-mail bezpośrednio z poziomu strony.
Możliwość walidacji danych wejściowych (sprawdzanie poprawności adresu e-mail, wymagane pola).
- **Przykład użycia:**
Potencjalny klient lub współpracownik wysyła zapytanie dotyczące projektów, wykorzystując formularz.

2.5. Integracja z Mediami Społecznościowymi

- **Opis funkcji:**
Ikony/linki do profili na LinkedIn, GitHub, Twitterze, Facebooku itp.
- **Przykład użycia:**
Użytkownik po kliknięciu w ikonę przechodzi do zewnętrznej strony z profilem autora.

2.6. (Opcjonalnie) Blog / Aktualności

- **Opis funkcji:**
Sekcja, w której autor może publikować krótkie artykuły lub notatki dotyczące swoich projektów, doświadczeń czy przemyśleń na temat branży.
- **Przykład użycia:**
Użytkownik przegląda najnowsze wpisy, które dodatkowo budują wizerunek eksperta.

3. Stos Technologiczny

3.1. Frontend

- **Framework: Next.js + TypeScript**
Zapewnia nowoczesne rozwiązania, wydajność oraz możliwość serwerowego renderowania (SSR) dla lepszej optymalizacji SEO.
- **UI Libraries:**
 - **shadcn/ui:** wykorzystaj przynajmniej jeden komponent
 - **Acternity:** integracja elementu graficznego np. animowanego przycisku lub karuzeli
 - **Magic UI:** wykorzystanie komponentu do formularzy lub interaktywnych widgetów
- **Style i RWD:**
 - **SASS/SCSS** lub **styled-components**
 - **Tailwind CSS** jako alternatywa dla szybkiego prototypowania (opcjonalnie)

3.2. Backend

- Jeśli strona jest statyczna lub typu "serverless", wystarczy wdrożenie na platformach wspierających Next.js (Vercel, Netlify).

- **Formularz kontaktowy:** Możliwość użycia prostego API (np. Next.js API routes) lub usługi typu ntfy.sh/mailer do obsługi wiadomości.

3.3. Hosting i Domeny

- **VPS:** Jeśli planujesz pełną kontrolę nad serwerem, możesz wykupić VPS, ale standardowo wystarczy platforma hostingowa wspierająca Next.js.
- **Domena:** Skonfiguruj domenę (np. portfolio.twojanazwa.pl) – upewnij się, że DNS jest poprawnie ustawiony.

4. Harmonogram Projektu

Tydzień 1: Planowanie i Projektowanie

- **Dzień 1-2:**
 - Określenie celu, zakresu i funkcjonalności strony.
 - Przygotowanie diagramów, makiet i szkiców UX/UI.
- **Dzień 3-4:**
 - Wybór technologii i konfiguracja środowiska deweloperskiego.
 - Utworzenie repozytorium kodu (np. na GitHub).
- **Dzień 5:**
 - Opracowanie wstępnej dokumentacji technicznej i planu projektu (takiego jak niniejszy).

Tydzień 2: Rozwój Frontendu

- **Dzień 1-2:**
 - Implementacja strony głównej i sekcji "O mnie".
 - Użycie komponentów z bibliotek shadcn/ui, Acternity, Magic UI.
- **Dzień 3-4:**
 - Budowa portfolio: galeria projektów, szczegóły projektu, integracja linków.
 - Prace nad responsywnością (RWD).
- **Dzień 5:**
 - Utworzenie szkieletu bloga lub sekcji aktualności (opcjonalnie).

Tydzień 3: Rozwój Backend i Integracja

- **Dzień 1:**
 - Implementacja API w Next.js do obsługi formularza kontaktowego (np. endpoint do wysyłki e-maili).
 - Konfiguracja bazy dla ewentualnego systemu komentarzy lub logów (opcjonalnie).
- **Dzień 2-3:**
 - Integracja logiki kontaktu: walidacja formularza, testy wysyłki wiadomości.
 - Wdrożenie integracji z zewnętrznymi serwisami (mediami społecznościowymi, ewentualne logi analityczne).
- **Dzień 4-5:**
 - Testy funkcjonalności – zarówno na etapie developmentu, jak i po integracji.
 - Dokonanie poprawek UX/UI w razie potrzeby.

Tydzień 4: Finalizacja, Testowanie i Wdrożenie

- **Dzień 1-2:**
 - Przeprowadzenie kompleksowych testów (testy manualne oraz, jeśli możliwe, automatyczne).
 - Testy responsywności na różnych urządzeniach oraz przeglądarkach.
- **Dzień 3:**
 - Końcowe poprawki, optymalizacja kodu i wydajności.
 - Finalizacja dokumentacji technicznej oraz instrukcji wdrożenia.
- **Dzień 4:**
 - Konfiguracja i wdrożenie aplikacji na docelowy serwer (Vercel, VPS lub inna platforma).
 - Testowanie dostępności pod domeną (konfiguracja DNS).
- **Dzień 5:**
 - Finalna prezentacja projektu i ewentualne demo (możliwość zaprezentowania portfolio dla potencjalnych klientów lub pracodawców).

5. Podział Ról i Organizacja Prac

Jeden deweloper:

W przypadku projektu jednoosobowego – wszystkie zadania realizowane są przez jedną osobę z zachowaniem samodzielności w zakresie planowania, programowania, testowania i wdrożenia.

W zespole (opcjonalnie):

Jeśli projekt realizowany jest przez zespół:

- **Project Manager:** Koordynacja prac, ustalanie priorytetów, zarządzanie repozytorium kodu i dokumentacją.
- **Frontend Developer:** Implementacja interfejsu użytkownika, integracja bibliotek UI, responsywność.
- **Backend Developer:** Tworzenie API, integracja formularza kontaktowego, konfiguracja serwera.
- **Tester/QA:** Testowanie aplikacji, zgłaszanie błędów i monitorowanie jakości końcowej.

6. Monitorowanie, Testowanie i Backupy

- **Monitorowanie:**
 - Użyj narzędzi takich jak Sentry lub LogRocket do monitorowania błędów aplikacji.
 - Monitorowanie wydajności za pomocą Lighthouse lub WebPageTest.
- **Testy:**
 - Testy jednostkowe (np. z użyciem Jest dla komponentów React/Next.js).
 - Testy integracyjne API.
- **Backup danych:**
 - Jeśli projekt korzysta z bazy danych (np. dla logów formularza), zaplanuj automatyczny backup bazy.

7. Wdrożenie i Publikacja

- **Konfiguracja domeny:**
 - Po wdrożeniu aplikacji skonfiguruj domenę (DNS, certyfikat SSL).
 - Upewnij się, że wszystkie ścieżki i linki działają poprawnie.
- **Publikacja:**
 - Udostępnij stronę na wybranej platformie hostingowej.
 - Przeprowadź testy dostępności (np. using Google Search Console) i w razie potrzeby wprowadź korekty.