**Sprawozdanie 1**

Projektowanie zorientowane na użytkownika

Grupa IO 2

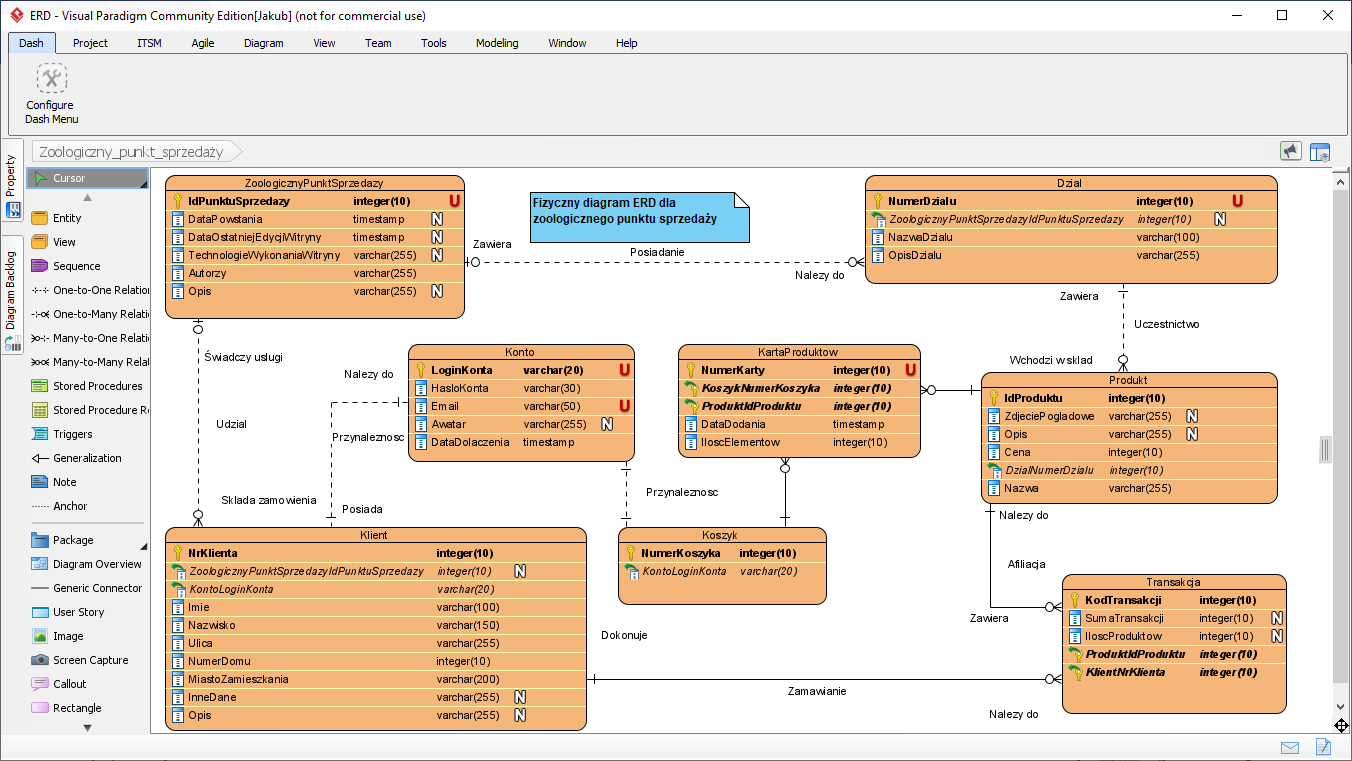
Jakub Krolik

Arkadiusz Jaszek

Marcel Niedziela

Zoologiczny sklep internetowy skierowany dla miłośników zwierząt. Witryna będzie posiadać przyjazny dla użytkownika jaki i administratora interfejs. 4 podstawowymi kategoriami, których produkty będą do kupienia w sklepie to zwierzęta, zabawki dla zwierząt, karma i inne. Wszystkie produkt, kategorie jak i zarejestrowani użytkownicy będą mieli swoje pozycje w bazie danych, której fizyczna wersja diagramu ERD widoczna jest poniżej. Baza wraz z upływem cyklu życia oprogramowania będzie rozwijana. Jej cechami są integracja danych, ich współdzielenie oraz bezpieczeństwo.

W końcowej fazie projektowania do systemu zaimplementowane zostaną testy e2e oraz m.in. rozwiązania chmurowe.

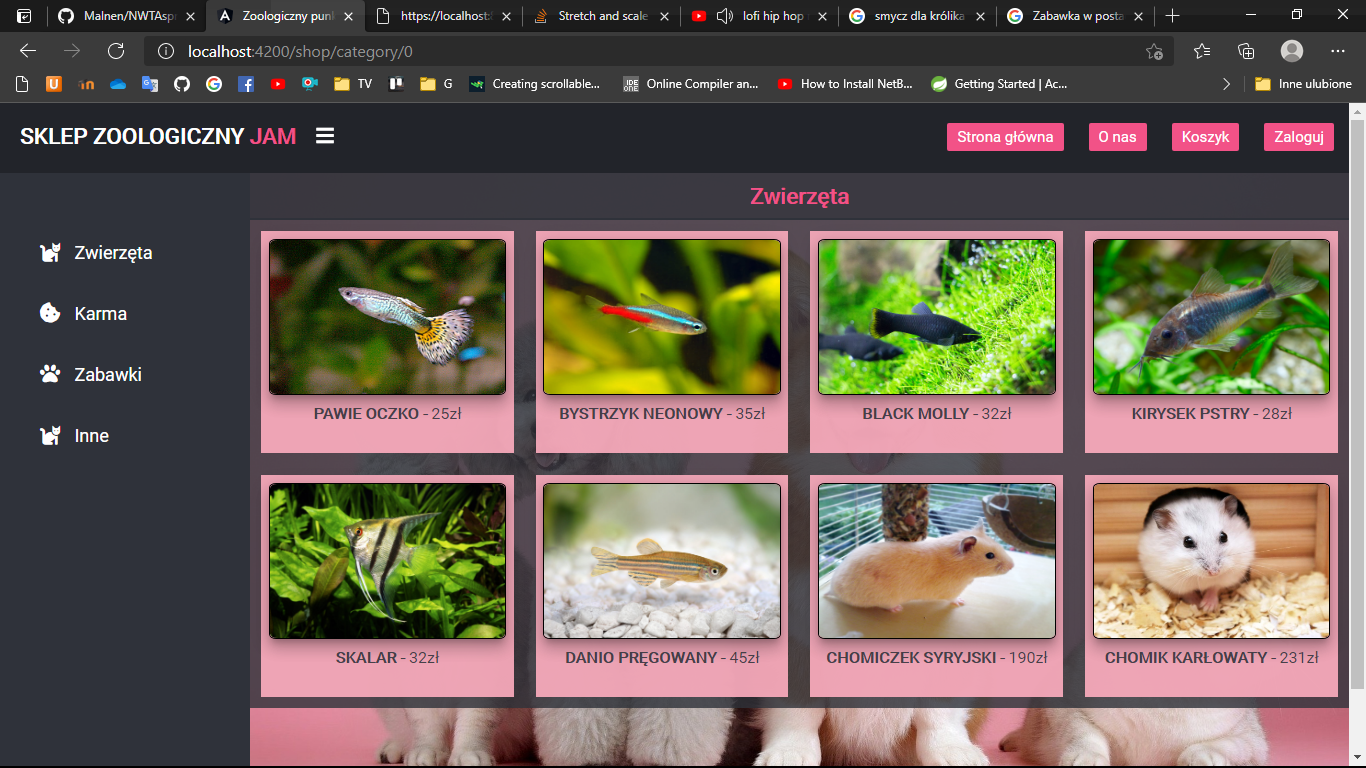


**Top features:**

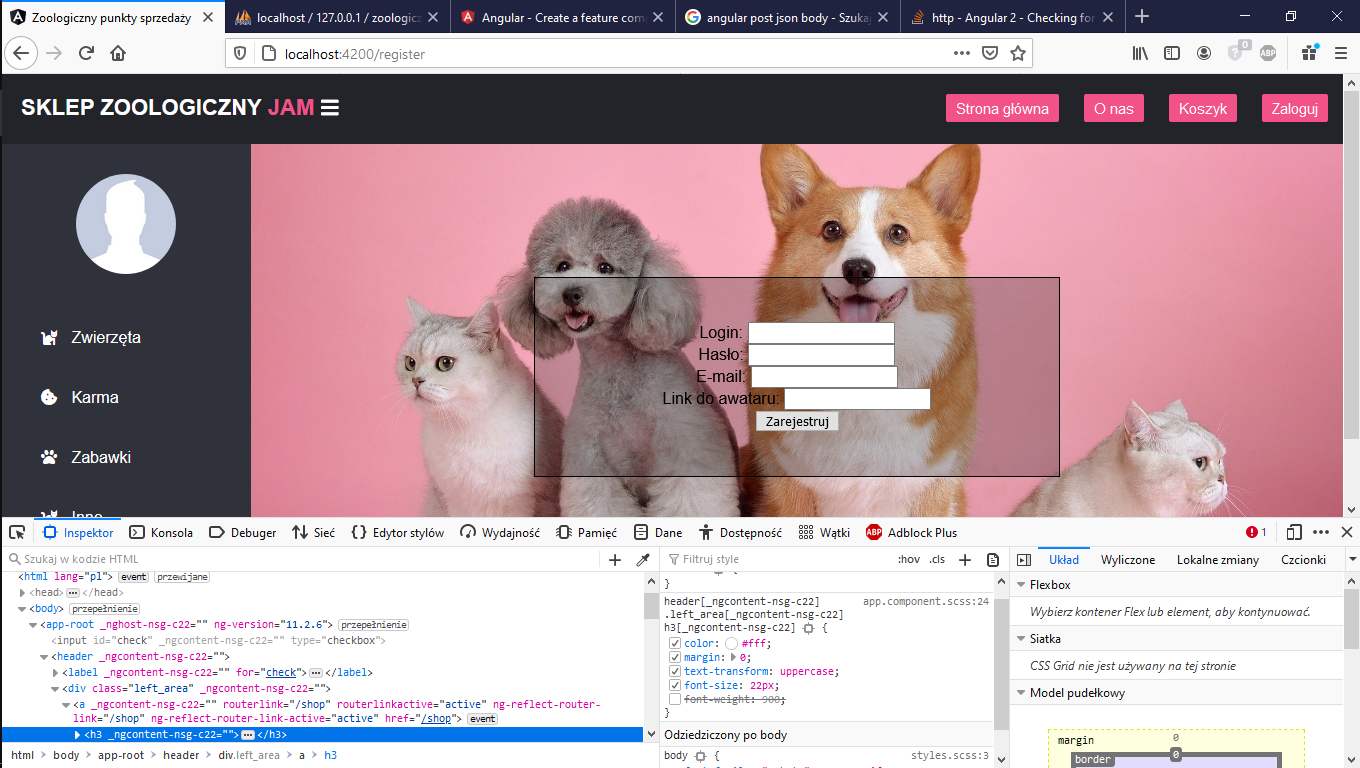
* System będzie umożliwiał logowanie i rejestracje nowych użytkowników
* System będzie umożliwiał dodawanie produktów do koszyka dla zalogowanych użytkowników
* System będzie prezentował opis i zdjęcie poglądowe produktów w sklepie
* W systemie produkty będą posegregowane w kategorie, dzięki którym użytkownik będzie mógł łatwo znaleźć produkt którego szuka
* System będzie zaimplementowany jako aplikacja webowa i będzie zawierał zoptymalizowaną wersję pod urządzenia mobilne
* Stworzona aplikacja będzie posiadała przyjazny dla użytkowników layout oraz nowoczesny interfejs użytkownika

**Poglądowe, wstępne wersje witryny:**

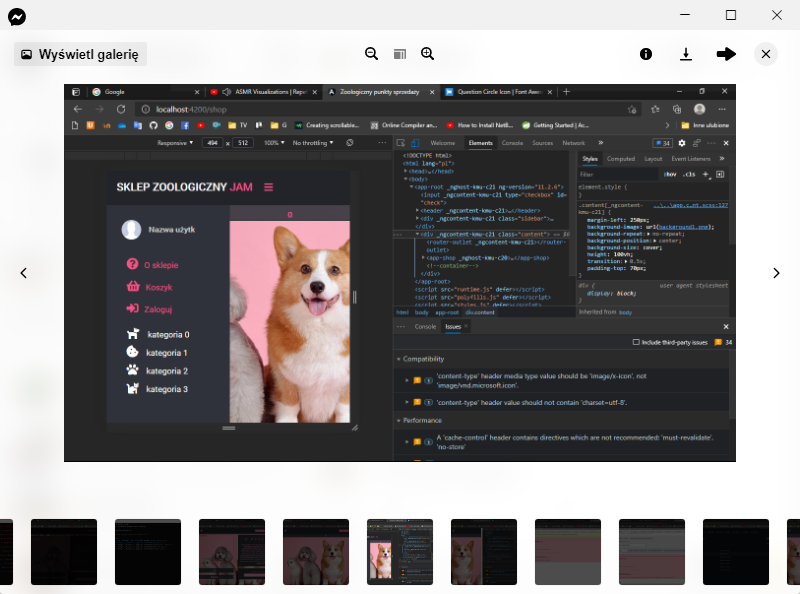
* Lewe menu



* Rejestracja użytkownika bez wygenerowanego arkusza stylów



* Widok menu sidebar w wersji mobilnej



**Ograniczenia w witrynie:**

* Użytkownik nie posiadający konta w bazie danych nie może dodać produktów do swojego koszyka
* Klient może przejrzeć na stronie tylko produkty mające swoje miejsce w bazie danych

**Potrzeby użytkownika, atrybuty oraz postulaty UX:**

Potrzeby użytkownika są badane na każdym procesie wytwarzania oprogramowania:

- podczas planowania

- podczas implementacji

- podczas testowania

Implementacja systemu od strony programistycznej odznacza się zapamiętywalnością. Stosowane są nazwy zmiennych oraz funkcji, których odtworzenie w czasie projektowania kolejnej wersji systemu będzie znacznie prostsze.

System staramy się projektować zgodnie z zasadą użyteczności. Zależy nam aby witryna była atrakcyjna i użyteczna. Dbamy o wygodną nawigację i prawidłowe nazewnictwo. W projekcie zostały zastosowane także specjalne ikony i grafika skojarzone z tematyką witryny.

**Lista użytkowników:**

* Użytkownik niezalogowany – oglądanie kategorii i produktów w sklepie
* Użytkownik zalogowany do sklepu – korzystanie z pełnych możliwości sklepu, dodawanie produktów do koszyka

**Jaki problem rozwiązuje projekt?**

Projekt pokazuje problem realizacji sklepu internetowego. Aby stworzyć serwis sklepowy potrzebne jest uruchomienie wiele narzędzi oraz wzajemne współgranie ich ze sobą; począwszy od front-endu, aby zachęcić użytkownika do częstych odwiedzin na stronie; poprzez redagowanie opisów przedmiotów oraz ustalanie cen. Możliwe, że najbardziej kluczowym elementem systemu jest jednak implementacja backendu opartego o rozwiązania bezpieczeństwa, implementacji integralnej bazy danych i połączeń między encjami. Problemem projektu jest właściwe współgranie wyżej wymienionych aspektów.

**Na ten czas witryna posiada:**

* Mechanizm rejestracji użytkownika, logowanie
* Możliwość przeglądania produktów
* Możliwość dodania produktów do koszyka
* Opracowaną wersję mobilną strony
* Rozwijany panel z kategoriami produktów w sklepie
* Opis dostępnego w sklepie produktu wraz z jego obrazkiem

**Użyte narzędzia oraz technologie:**

* Spring
* Angular
* HTML/SCSS
* Postman
* MySql
* Użycie serwisów internetowych: Pixabay, fontawesome, colorhunt

**Podział zadań w grupie:**

Arek Jaszek – frontend

(JS, Angular, struktura klas, połączenie z bazą, mechanizm rejestracji i logowania, koszyk)

Jakub Krolik – full-stack

(HTML, SCSS, wersja mobilna, baza mysql, redagowanie opisów)

Marcel Niedziela – backend

(JAVA spring, postman, struktura klas, endpointy, połączenie z bazą, obsługa błędów, chmura)