

1. Aukcje NFT .....	2
1.1 Wstępna wizja projektu .....	2
1.2 Dokumentacja .....	2
1.2.1 Specyfikacja Wymagań .....	2
1.2.1.1 Plan pracy .....	3
1.2.2 Definicja Architektury .....	3
1.2.2.1 Wirtualizacja i konteneryzacja .....	4
1.2.3 Specyfikacja Analityczna .....	4
1.2.4 Specyfikacja Projektowa .....	7
1.2.5 Specyfikacja Testów .....	10
1.2.6 Podręcznik Użytkownika .....	13
1.2.7 Podręcznik Administratora .....	16
1.3 UX/UI .....	17
1.3.1 Aktorzy .....	17
1.3.2 Persony .....	17
1.3.3 User Stories .....	17
1.3.4 Przypadki użycia .....	18
1.3.5 Wireflow .....	21
1.3.6 Ekrany .....	21
1.3.7 Domenowy słownik pojęć .....	23
1.4 Modelowanie Zagrożeń .....	23

# Aukcje NFT

---

Wstępna wizja projektu

## 1 Cel projektu

Celem projektu jest stworzenie prototypu aplikacji umożliwiającej tworzenie tokenów NFT i ich sprzedaż w ramach aukcji prowadzonych przez szkoły. Aplikacja ma realizować 3 podstawowe funkcjonalności:

- zakładanie konta/portfela
- mintowanie NFT
- przeprowadzanie aukcji NFT

### 1.1 Cele biznesowe

Licytacje NFT są istotnym elementem budżetu liceów w Kalifornii. Aplikacja ma na celu ułatwienie przeprowadzania aukcji i zautomatyzowanie niektórych rzeczy.

### 1.2 Cele użytkowe

Aplikacja powinna umożliwiać użytkownikom w proste tworzenie własnych NFT i wystawianie ich na aukcje przeprowadzane w jak najbardziej intuicyjny sposób.

## 2 Wstępna wizja projektu

W ramach projektu tworzona będzie aplikacja o architekturze klient-serwer. Aplikacja będzie umożliwiać szybkie założenie konta poprzez inne serwisy takie jak google/facebook. Po zalogowaniu na stronie głównej wyświetlać się będą trwające i nadchodzące w najbliższym czasie aukcje, w których użytkownik może uczestniczyć. Po wygraniu aukcji, token NFT zostaje powiązany z kontem i dodawany do repozytorium w profilu użytkownika, gdzie może być oglądany lub ponownie wystawiany na aukcje. Aplikacja będzie miała własny portfel, z którego pobierane będą środki na zakup tokenów NFT i który może być doładowywany przez profil użytkownika.

Dokumentacja

Specyfikacja Wymagań

### Wymagania w konwencji MoSCoW

Dodatkowy podział na 3 typy wymagań:

- **niebieski** - wymagania biznesowe
- **zielony** - wymagania użytkowe
- **pomarańczowy** - wymagania systemowe
- **czerwony** - wymagania pasujące do wielu z powyższych typów

### Must

- **zapisywanie NFT w portfelu**
- **po wygraniu aukcji token wiązany z kontem (bezpośrednia realizacja przez aplikację)**
- **łatwe tworzenie i branie udziału w aukcji**
- **użytkownicy nie za bardzo muszą myśleć jak to się dzieje**
- **szynkowe założenie konta (np. logowanie za pomocą konta Google/Facebook)**
- **użytkownik nie musi tworzyć zewnętrznego portfela NFT, portfel w naszej aplikacji jest tworzony wraz z kontem**
- **na stronie głównej trwające lub zbliżające się aukcje**
- **możliwość przeglądania naszych NFT (jako forma kolekcjonowania)**
- **własny portfel (z walutą) z którego będą pobierane środki na opłacenie aukcji (suma kwot postawionych przez użytkownika we wszystkich aukcjach nie może przekraczać kwoty w portfelu)**

## Should

- możliwość wystawiania ofert typu "kup teraz" przez użytkowników
- kolekcje/grupy NFT (np. powiązane tematycznie lub odgórny podział na kilka klas)

## Could

- możliwość tworzenia aukcji przez użytkowników
- bardziej złożone smart-contracty, gdzie szkoła zarabia na obrocie wtórnym
- system osiągnięć (np. ilość posiadanych NFT)

## Won't

- utrata dostępu do pliku po jego sprzedaży (trudne technicznie)

### Przypadki użycia

Szkoła wystawia na aukcje dzieła tworzone przez uczniów/pracowników, w celu finansowania swojej działalności. Zainteresowane osoby mogą brać udział w licytacjach.

#### User Stories

#### Plan pracy

Przewidywany termin zakończenia	Funkcjonalności
3.12.22	<ul style="list-style-type: none"> <li>• szybkie założenie konta (np. logowanie za pomocą konta Google/Facebook)</li> <li>• na stronie głównej trwające lub zbliżające się aukcje</li> <li>• możliwość przeglądania naszych NFT (jako forma kolekcjonowania)</li> </ul>
17.12.22	<ul style="list-style-type: none"> <li>• zapisywanie NFT w portfelu</li> <li>• łatwe tworzenie i branie udziału w aukcji</li> </ul>
5.01.22	<ul style="list-style-type: none"> <li>• stworzenie własnego portfela (z walutą) z którego będą pobierane środki na opłacenie aukcji (suma kwot postawionych przez użytkownika we wszystkich aukcjach nie może przekraczać kwoty w portfelu)</li> </ul>
15.01.22	<ul style="list-style-type: none"> <li>• możliwość wystawiania ofert typu "kup teraz" przez użytkowników</li> <li>• kolekcje/grupy NFT (np. powiązane tematycznie lub odgórny podział na kilka klas)</li> </ul>
opcjonalnie na koniec projektu	<ul style="list-style-type: none"> <li>• możliwość tworzenia aukcji przez użytkowników</li> <li>• bardziej złożone smart-contracty, gdzie szkoła zarabia na obrocie wtórnym</li> <li>• system osiągnięć (np. ilość posiadanych NFT)</li> </ul>

### Definicja Architektury

#### Definicja architektury:

##### Diagram C4

Szablony architektoniczne:

Ze względu na strukturę API, będziemy korzystali z wzorca MVC, który będzie działał w sposób następujący:

- Program poprzez kontroler wybiera dane wejściowe od użytkownika i za ich pomocą wybiera stosowny model
- Model za pomocą wejścia od użytkownika wybiera odpowiedni widok, który powinien zostać wyświetlony, a także zapewnia mu dane
- Widok przedstawia dane, które zwrócił model w sposób czytelny dla użytkownika

Elementy struktury i ich interfejsy:

Django implementuje widoki, które spełniają rolę kontrolerów w modelu MVC. Za ich pomocą będziemy obsługiwać wyżej wymieniony model. Następnie backend wybierze odpowiednio wygenerowany model, który pozwoli na przetworzenie danych pobranych z bazy. Po przetworzeniu danych zwrócony zostanie odpowiedni widok.

#### **Interakcje pomiędzy elementami:**

Program opierał się będzie na wykorzystaniu architektury trójwarstwowej, wykorzystującej API oraz integrację z bazą danych za pomocą (tu wstawić). Każde żądanie, wysypane przez klienta za pomocą GUI będzie przetwarzane przy pomocy serwera, który pobierał będzie odpowiednie składowe z bazy danych. Istotne jest to, że logowanie i autoryzowanie użytkownika odbywać się będzie przy pomocy Firebase, który pozwoli na skuteczne wyeliminowanie problemów związanych z bezpieczeństwem.

#### **Przyjęte rozwiązania:**

Po konsultacjach z dr hab. inż. Mariuszem Kaletą zdecydowaliśmy się przyjąć sposób postępowania, który będzie bardzo intuicyjny i łatwy do przyswojenia dla wszystkich użytkowników, ze względu na fakt, że nasz system nastawiony jest na osoby dorosłe, a także starsze, które mogą mieć problem z bardziej zaawansowanymi czynnościami. Kluczowe jest minimalizowanie liczby trudności, które potencjalny klient mógłby napotkać, przy jednoczesnym zachowaniu wszystkich funkcjonalności. Z tego też powodu zdecydowaliśmy się wprowadzić następujące ułatwienia:

- Umożliwienie użytkownikom logowania się za pomocą konta Google/Facebook
- Automatyczne zakładanie portfela, który przechowywałby środki pieniężne, wpłacane uprzednio, aby za pomocą tych środków brać udział w aukcjach
- Wyświetlanie najpopularniejszych aukcji w widocznym miejscu, aby klient nie musiał przechodzić przez ściany tekstu w poszukiwaniu aukcji
- Możliwość łatwego przeglądania NFT

#### **Mechanizmy techniczne:**

- Ponieważ nasz program będzie obsługiwał zapytania do serwera, ten serwer spoczywał będzie w chmurze, dostępnej publicznie, do której dostęp będzie miał każdy, za pomocą dowolnej przeglądarki internetowej
- Wybrana chmura to Microsoft Azure, ze względu na darmowe środki dla studentów
- Do przechowywania danych wybraliśmy bazę danych Oracle, ze względu na darmowe licencje dla studentów, natomiast uwierzytelnienie i autoryzacja odbywać się będzie za pomocą Firebase
- Wykorzystamy Bitbucket CI, aby cały czas zapewniać stabilną wersję naszego programu.
- Warstwa graficzna naszego programu zostanie napisana w React, backend opierał się będzie na Pythonie, natomiast rolę połączenia między nimi spełni Django.

#### **Wirtualizacja i konteneryzacja**

##### **Wirtualizacja i konteneryzacja:**

Do realizacji projektu postawiliśmy maszynę wirtualną na platformie Azure, na której jest zdeployowany backend oraz frontend portalu na zdefiniowanych przez nas portach (8000 backend oraz 3000 frontend). Do zarządzania maszyną mają dostęp jedynie osoby posiadające klucz ssh, natomiast dostęp do naszej zdeployowanej aplikacji ma każdy (o ile maszyna wirtualna jest uruchomiona).

Na maszynie wirtualnej chodzą dwa kontenery Docker, jeden z frontendem a drugi z backendem aplikacji. Użytkownik łączy się do frontendu aplikacji, który pod spodem wywołuje odpowiednio przygotowane metody HTTP do kontenera z backendem, który wysyła mu odpowiedzi. Są one następnie prezentowane użytkownikowi. (Rozdzielenie klienta od serwera. Klient nie wie nic o logice pod spodem, serwer nie wie jak prezentowane są dane.)

#### **Zidentyfikowane zagrożenia:**

Uwierzytelnianie użytkownika – pozbylismy się problemu przechowywania danych wrażliwych poprzez oddelegowanie uwierzytelniania na usługi Google Firebase, co pozwoliło zaoszczędzić dużo czasu realizacji projektu.

Wywoływanie metod API z zewnątrz – ustawiemy komunikację z backendem możliwą jedynie ze strony frontendu, wywołania endpointów bezpośrednie z dowolnego adresu IP nie będą możliwe.

Wywołanie metod API, do których użytkownik nie ma dostępu np. chcę sprzedać nie swojego NFT – poprzez metodę POST będzie przekazywany identyfikator aktualnie zalogowanego użytkownika, backend będzie sprawdzał czy zalogowany użytkownik ma prawo do zasobu do jakiego próbuje się odwołać.

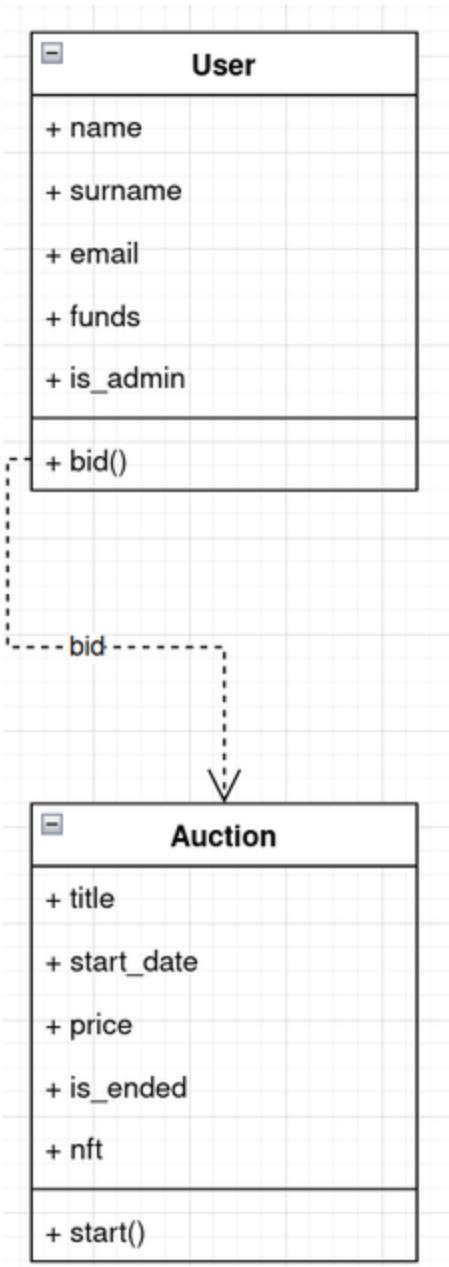
#### **Specyfikacja Analityczna**

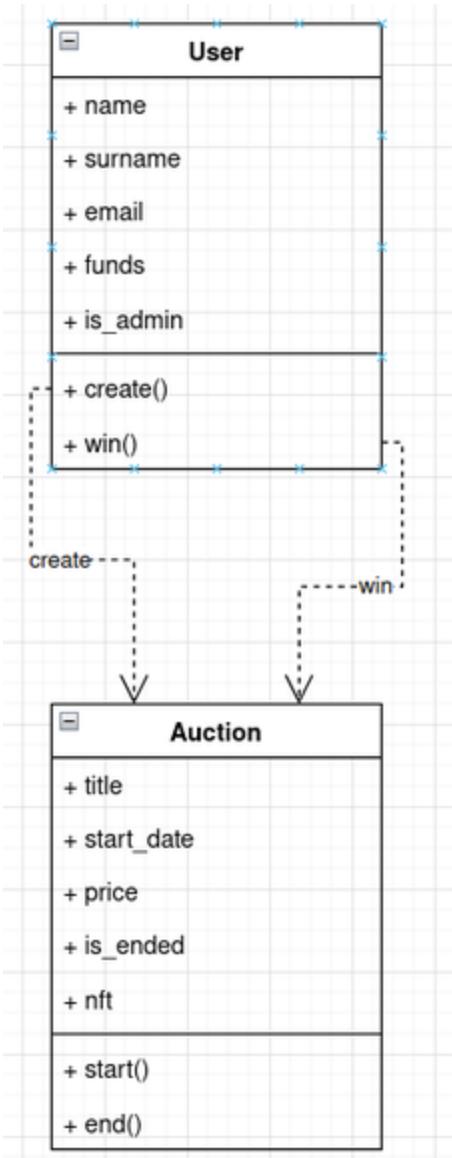
##### **Słownik Pojęć**

##### **Domenowy słownik pojęć**

##### **Model dziedzinowy**







Przypadki użycia

[Przypadki użycia](#)

Specyfikacja Projektowa

### Języki programowania i frameworki

Python → Django

JavaScript → React

### RestAPI Backend

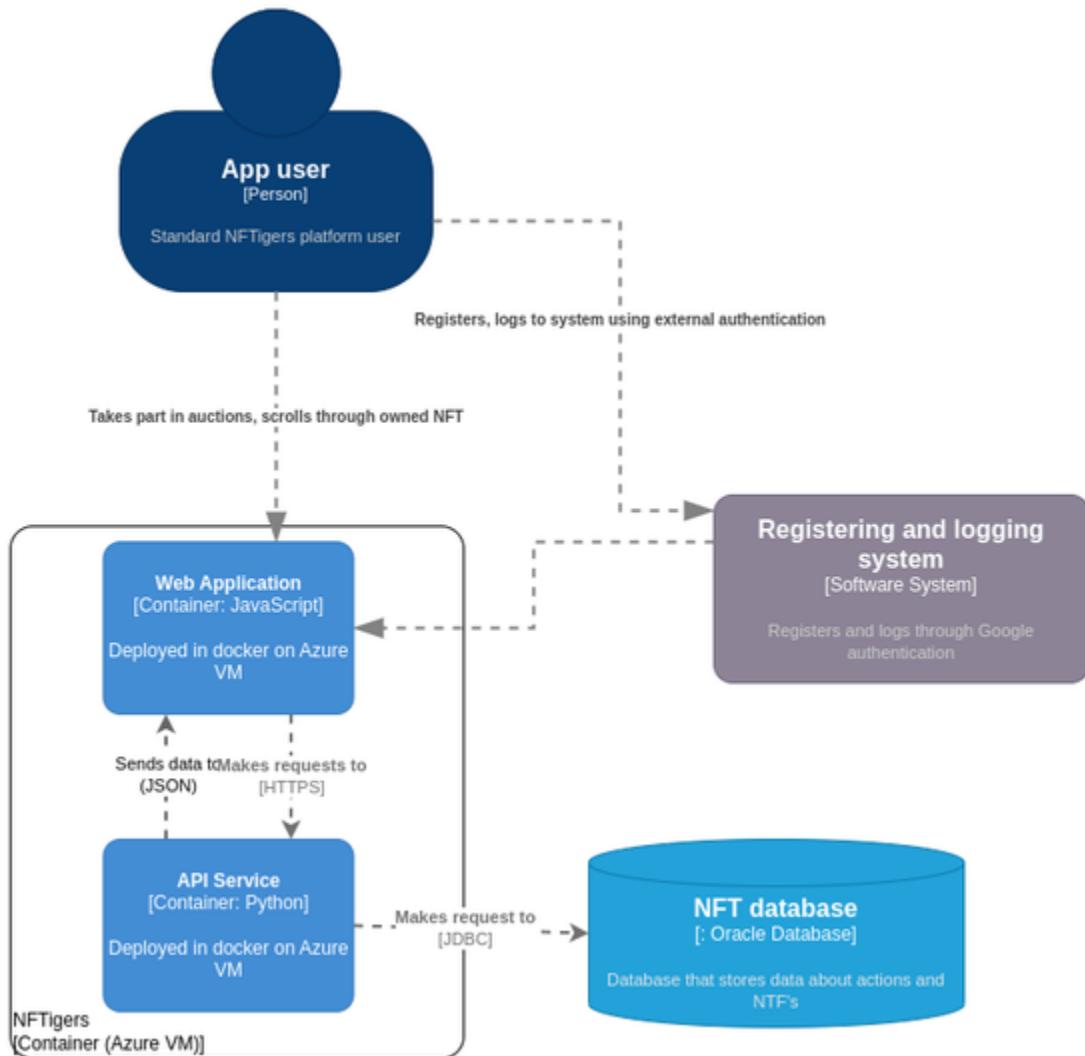
```

GET api/auctions/?class={class_id}&category={category_id}
POST api/auctions/
POST api/nfts/
POST api/wallet/
PUT api/wallet/
POST api/users/
GET api/classes/
GET api/categories/

```

1. zwraca listę aukcji które się jeszcze nie zakończyły
2. podbijanie aukcji (oczuje w body: usId - id użytkownika, nftId - id aukcji, bid - kwota)
3. pobieranie nft, które posiada użytkownik (oczuje w body: usId)
4. pobieranie portfela użytkownika (oczuje w body: usId)
5. aktualizacja portfela użytkownika (oczuje w body: usId, income)
6. rejestracja użytkownika w naszej bazie (oczuje w body: method - czy google/facebook, usId, email)
7. zwraca listę klas w szkole
8. zwraca listę kategorii pracowników

#### Diagram struktury systemu



## Błędy

Backend aplikacji w razie błędnych wywołań zwraca odpowiednie statusy HTTP, które obługuje frontend, wyświetlając odpowiednie komunikaty użytkownikowi.

## Baza danych (Oracle):

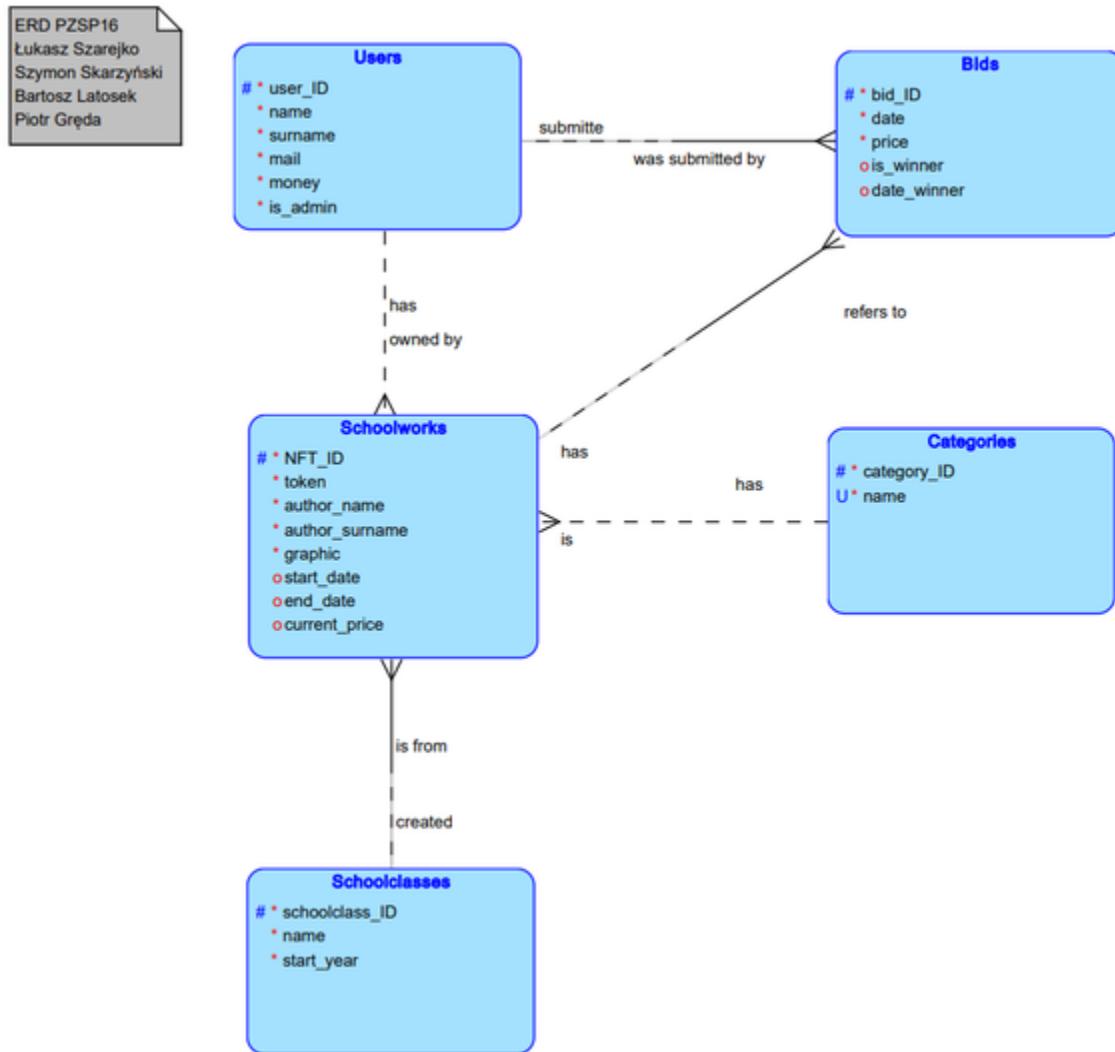
### Planowany sposób implementacji:

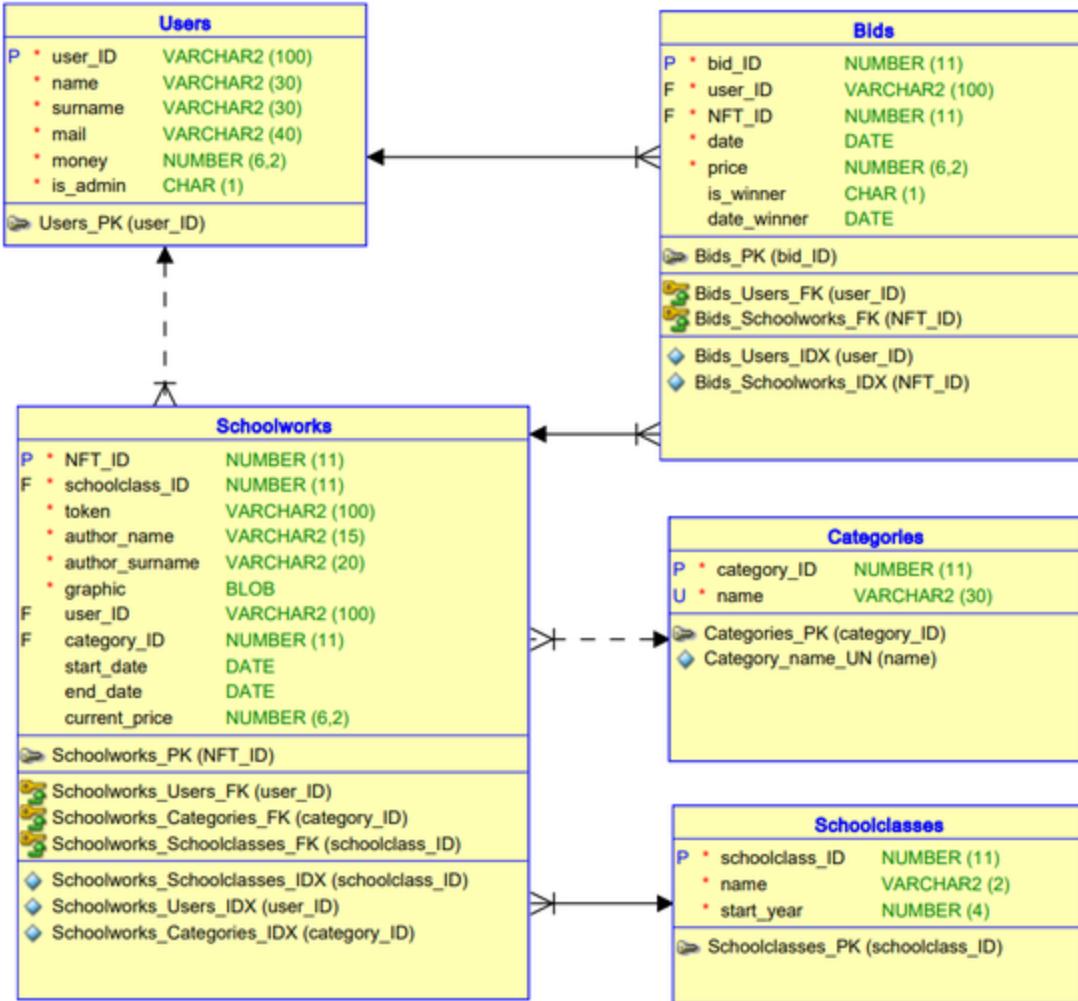
Z racji na wykorzystanie Django w projekcie, wykonamy bazę danych w oparciu o podejście code-first. W projekcie zostaną utworzone klasy (modele), które przy pomocy framework'u [Django będą migrowane](#) do bazy danych Oracle udostępnionej przez uczelnię. Dostęp do danych z bazy będzie następował poprzez wykorzystanie [Django QuerySet API](#).

Model koncepcyjny został wykonany jeszcze przed implementacją klas w kodzie, dlatego może on ulec zmianie. Warto też dodać, że do logowania użytkownika użyjemy rozwiązań z Firebase, czyli logowania za pomocą konta facebook lub google i przechowywania danych przez Firebase. W związku z tym dane użytkownika takie jak login czy hasło nie będą przechowywane w naszej bazie oracle, czyli nie będzie ich w modelu.

Nie jest planowane programowanie w bazie danych.

### Logiczny oraz relacyjny model danych (po zmianach zasugerowanych przez prof. Traczyka):





Firebase

Do zarządzania logowaniem i rejestracją dane zostały wydzielone do oddzielnej bazy firebase.

## Makiety i Wireflowy

Wireflow

Ekrany

Specyfikacja Testów

- Testy automatyczne
- Miary jakości testów
- Testy systemowe
  - Rejestracja na platformie
  - Przeglądanie aktywnych lub nadchodzących aukcji
  - Przeglądanie posiadanych NFT
  - Branie udziału w aukcji
  - Wygranie aukcji
  - Dofładowanie portfela
  - Dodawanie aukcji przez administratora
  - Dodawanie kategorii przez administratora
  - Dodawanie klasy przez administratora

## Testy automatyczne

Testy automatyczne, w skład których wchodzą testy jednostkowe, integracyjne, modeli oraz API zostały wykonane przy pomocy frameworka Django, który udostępnia bardzo sprawny mechanizm do testowania. Testy poszczególnych aplikacji zostały umieszczone w folderze tests w folderach danych aplikacji np.

src/Django/NFTigers/users/tests/ Do ich wykonania należy użyć komendy **python manage.py test** będąc w odpowiednim folderze lub użyć skryptu **./scripts/test.sh** będąc w folderze głównym kodu.

## Miary jakości testów

W celu zbadania miary jakości takiej jak pokrycie kodu testami, użyliśmy biblioteki coverage. W celu sprawdzenia pokrycia wystarczy wykonać przygotowany skrypt będąc w głównych folderze repozytorium **.scripts/test.sh**

## Testy systemowe

Poniżej znajdują się scenariusze testów systemowych

### Rejestracja na platformie

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Przejdź na stronę rejestracji  
REZULTAT Strona rejestracji załadowała się.
- c. AKCJA Zarejestruj się  
REZULTAT Użytkownik został zarejestrowany i przekierowany na stronę logowania.
- d. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- e. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

### Przeglądanie aktywnych lub nadchodzących aukcji

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- c. AKCJA Przejdź na zakładkę aukcje  
REZULTAT Wyświetlona jest lista aktywnych lub nadchodzących aukcji.
- d. AKCJA Wybierz aukcję  
REZULTAT Wyświetlony jest opis aukcji, wraz z aktualną ceną.
- e. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

### Przeglądanie posiadanych NFT

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- c. AKCJA Przejdź na posiadane NFT  
REZULTAT Wyświetlona jest lista NFT posiadanych przez użytkownika.
- d. AKCJA Wybierz posiadane NFT.  
REZULTAT Wyświetlony jest opis NFT.
- e. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

### Branie udziału w aukcji

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- c. AKCJA Przejdź na zakładkę aukcje  
REZULTAT Wyświetlona jest lista aktywnych lub nadchodzących aukcji.

- d. AKCJA Wybierz aukcję  
REZULTAT Wyświetlony jest opis aukcji, wraz z aktualną ceną.
- e. AKCJA Wprowadź kwotę wyższą niż aktualna cena i zatwierdź  
REZULTAT Wyświetlony jest dialog zatwierdzający podbicie aukcji.
- f. AKCJA Zatwierdź podbicie  
REZULTAT Okno dialogowe znika, aktualna cena aukcji zmienia się na wprowadzoną kwotę.
- g. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

#### Wygranie aukcji

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- c. AKCJA Przejdź na zakładkę aukcję  
REZULTAT Wyświetlona jest lista aktywnych lub nadchodzących aukcji.
- d. AKCJA Wybierz aukcję  
REZULTAT Wyświetlony jest opis aukcji, wraz z aktualną ceną.
- e. AKCJA Wprowadź kwotę wyższą niż aktualna cena i zatwierdź  
REZULTAT Wyświetlony jest dialog zatwierdzający podbicie aukcji.
- f. AKCJA Zatwierdź podbicie  
REZULTAT Okno dialogowe znika, aktualna cena aukcji zmienia się na wprowadzoną kwotę.
- g. AKCJA Poczekaj na zakończenie aukcji (wracając do e. jeśli ktoś postawi wyższą kwotę niż twoja)  
REZULTAT NFT zostaje przypisane do użytkownika, a aukcja nie jest już możliwa do podglądu, przekierowanie użytkownika do strony głównej.
- h. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

#### Działanie portfela

- a. AKCJA Uruchom stronę internetową NFTigers  
REZULTAT Strona logowania załadowała się.
- b. AKCJA Zaloguj się w systemie  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie głównej.
- c. AKCJA Wybierz przycisk doładowania portfela  
REZULTAT Wyświetlone zostaje okienko dialogowe do doładowania.
- d. AKCJA Wprowadź kwotę i zatwierdź  
REZULTAT Okno znika, wartość portfela ulega zmianie.
- e. AKCJA Wyloguj użytkownika  
REZULTAT Użytkownik zostaje wylogowany.

#### Dodawanie aukcji przy administratorze

- a. AKCJA Uruchom stronę administracyjną NFTigers  
REZULTAT Strona logowania administratora załadowała się.
- b. AKCJA Zaloguj się w systemie administracyjnym  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie administracyjnej.
- c. AKCJA Wybierz przycisk nft po lewej stronie  
REZULTAT Wyświetlone zostają wszystkie aukcje w systemie.
- d. AKCJA Wybierz przycisk dodania nft  
REZULTAT Wyświetlona zostaje strona do dodania pracy ucznia.
- d. AKCJA Uzupełnij dane i zatwierdź  
REZULTAT Praca została zapisana jako nft i zostaje wyświetlona lista wszystkich aukcji z punktu c.
- e. AKCJA Wyloguj użytkownika z panelu administracyjnego  
REZULTAT Użytkownik zostaje wylogowany.

#### Dodawanie kategorii przez administratora

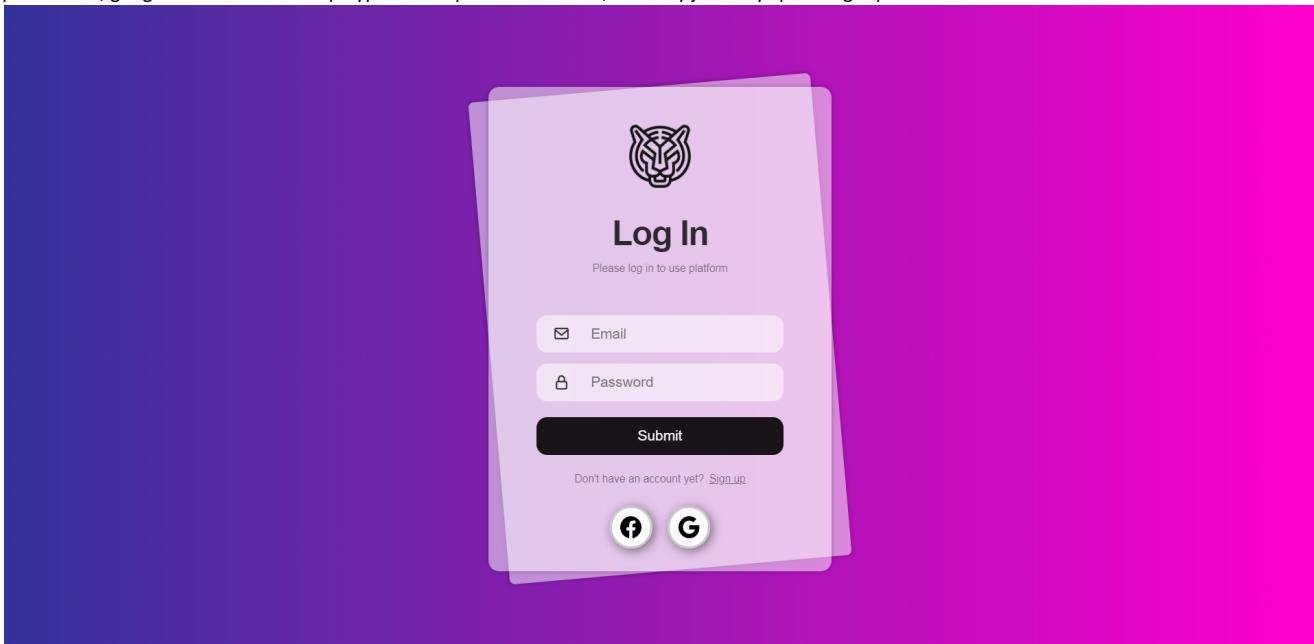
- a. AKCJA Uruchom stronę administracyjną NFTigers  
REZULTAT Strona logowania administratora załadowała się.
- b. AKCJA Zaloguj się w systemie administracyjnym  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie administracyjnej.
- c. AKCJA Wybierz przycisk category po lewej stronie  
REZULTAT Wyświetlone zostają wszystkie kategorie prac w systemie.
- d. AKCJA Wybierz przycisk dodania kategorii  
REZULTAT Wyświetlona zostaje strona do dodania kategorii NFT.
- e. AKCJA Uzupełnij dane i zatwierdź  
REZULTAT Kategoria zostaje zapisana i zostaje wyświetlona lista wszystkich kategorii z punktu c.
- f. AKCJA Wyloguj użytkownika z panelu administracyjnego  
REZULTAT Użytkownik zostaje wylogowany.

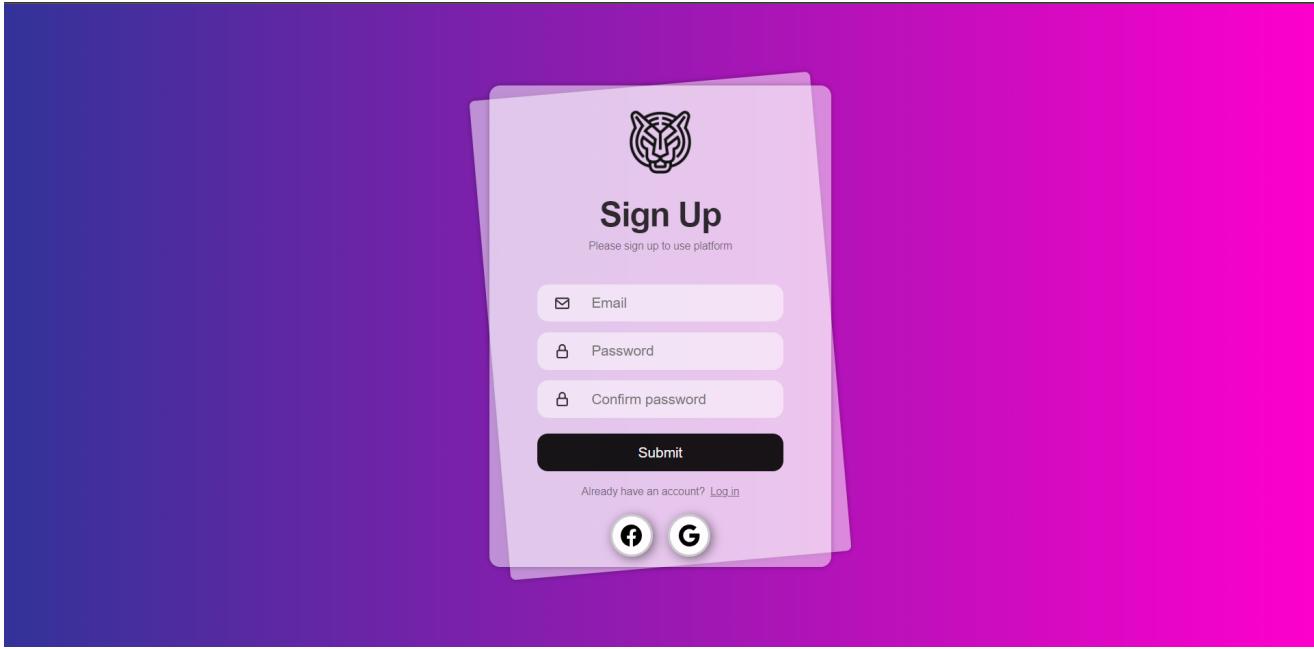
#### Dodawanie klasy przez administratora

- a. AKCJA Uruchom stronę administracyjną NFTigers  
REZULTAT Strona logowania administratora załadowała się.
- b. AKCJA Zaloguj się w systemie administracyjnym  
REZULTAT Użytkownik zalogował się w systemie i znalazł się na stronie administracyjnej.
- c. AKCJA Wybierz przycisk class po lewej stronie  
REZULTAT Wyświetlone zostają wszystkie klasy ze szkoły w systemie.
- d. AKCJA Wybierz przycisk dodania klasy  
REZULTAT Wyświetlona zostaje strona do dodania klasy szkolnej.
- e. AKCJA Uzupełnij dane i zatwierdź  
REZULTAT Klasa zostaje dodana i zostaje wyświetlona lista wszystkich klas z punktu c.
- f. AKCJA Wyloguj użytkownika z panelu administracyjnego  
REZULTAT Użytkownik zostaje wylogowany.

#### Podręcznik Użytkownika

1. Pierwszym krokiem do korzystania z aplikacji jest utworzenie konta w serwisie za pomocą ekranu rejestracji. Zakładamy 3 różne metody logowania: przez email, google oraz facebook. W przypadku nie posiadania konta, możemy je założyć przez signup.

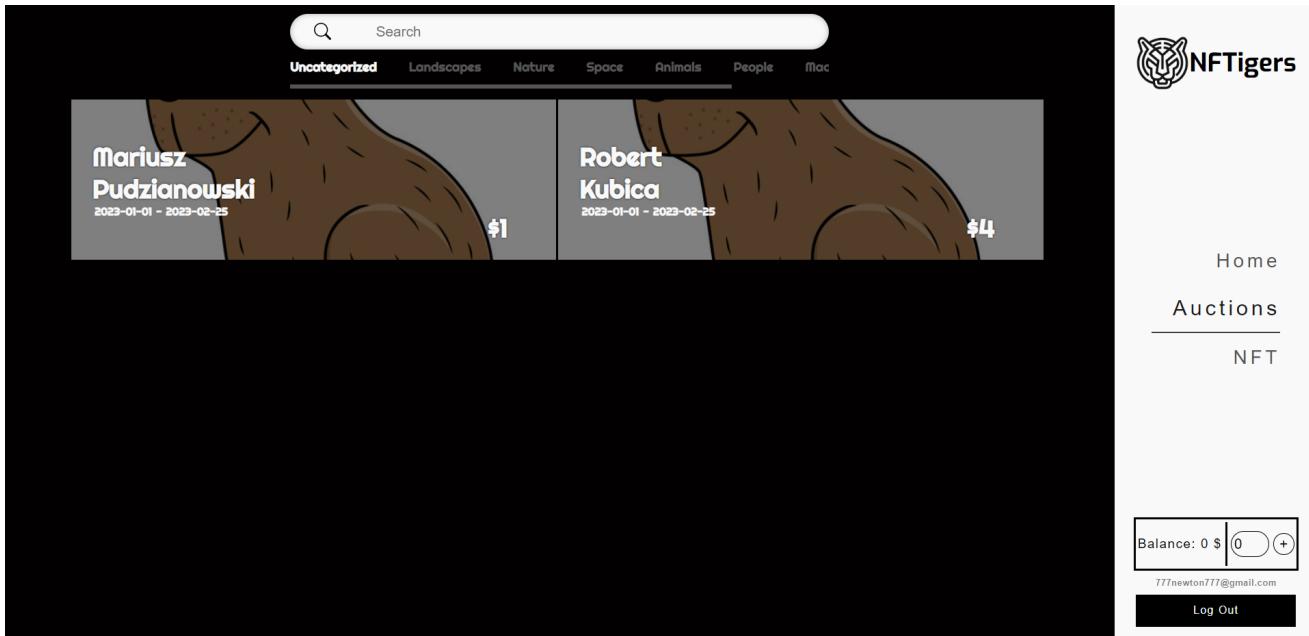




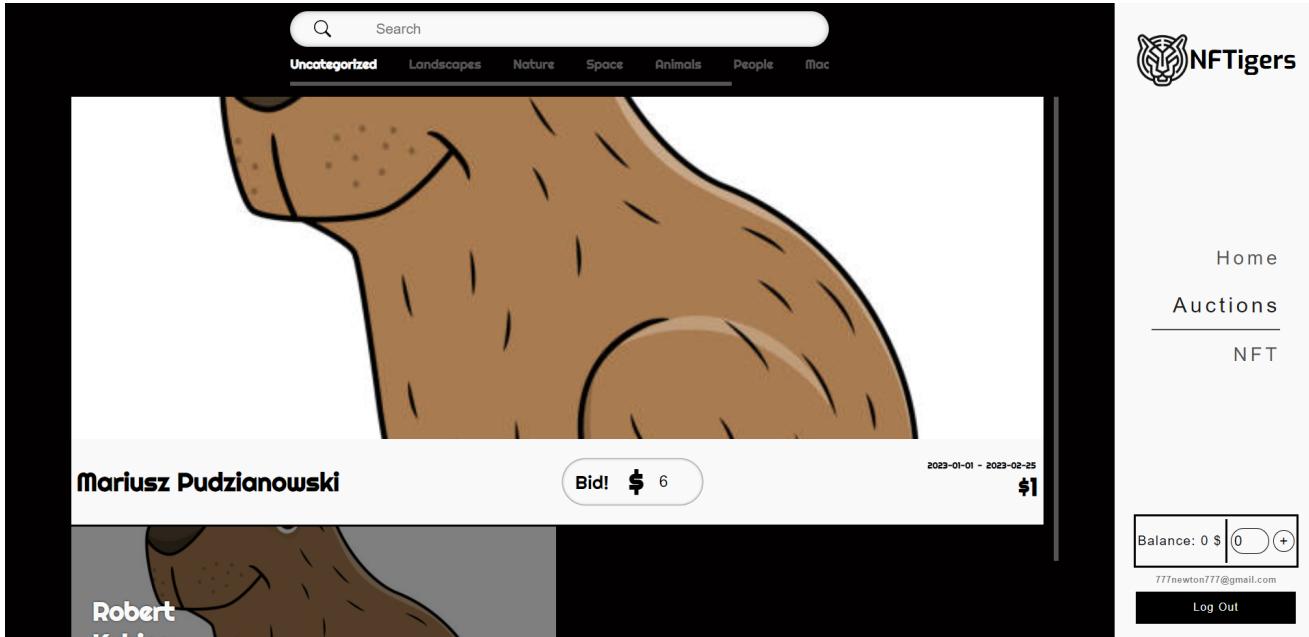
2. Po zalogowaniu zostajemy przeniesieni na stronę główną serwisu.

A screenshot of the NFTigers platform's home screen. The background is black. On the left side, there is a large, stylized logo consisting of the letters "NFTIGERS" in white and pink. The "N" is white with a black outline, and the "FTIGERS" part is pink with a black outline. Below this logo is a blue rectangular button with the text "Go To Auctions" in white. On the right side, there is a sidebar with a light gray background. At the top of the sidebar is the NFTigers logo. Below the logo are three menu items: "Home" (underlined), "Auctions", and "NFT". At the bottom of the sidebar, there is a balance indicator showing "Balance: 0 \$" next to a small input field containing "0" with a plus sign, and an email address "777newton777@gmail.com". A "Log Out" button is located at the very bottom of the sidebar.

3. Ze strony głównej serwisu, przy pomocy paska adresu po prawej stronie interfejsu możemy przejść do dostępnych aukcji.



4. Po wybraniu interesującej nas aukcji możemy podbić aktualną cenę, wpisując nową kwotę zakładu i klikając w przycisk "Bid".



5. Po zakończeniu aukcji, nft zostaje na stałe przypisane do naszego konta i może być przeglądane po przejściu do zakładki NFT's po prawej stronie interfejsu.

**Mariusz Pudzianowski**

2023-01-09  
\$5

NFTigers

- Home
- Auctions
- NFT**

Balance: 3 \$ |   777newton777@gmail.com

Log Out

6. Możemy przeglądać zdobyte na aukcjach nft, wraz z informacjami o autorze, cenie ostatecznej i dacie zakończenia aukcji.

#### Podręcznik Administratora

- [Budowa systemu z kodu źródłowego:](#)
- [Zbudowanie systemu z kodu źródłowego przy pomocy Docker'a:](#)
- [Konfiguracja systemu:](#)
- [Dodatkowe uprawnienia administratora](#)
- [Instrukcja aktualizacji oprogramowania:](#)
- [Tworzenie kopii zapasowych i odtwarzanie systemu:](#)

Budowa systemu z kodu źródłowego:

Ponieważ nasz system działa w środowisku [chmurowym](#) (domyślnie jest wyłączony, aby nie generować kosztów związanych z działającą chmurą) nie zaleca się budowania go lokalnie. Jeżeli jednak ktoś zdecyduje się na ten krok, poniżej opisane są propozycje.

Budowanie systemu zostało zautomatyzowane przy pomocy Docker'a. Bez niego ten proces jest znacznie dłuższy i bardziej skomplikowany, więc nie będzie zawarty w podręczniku. Jeżeli oczywiście zaistnieje potrzeba budowy systemu bez pomocy Docker'a, należy przeczytać pliki Dockerfile z obu repozytoriów, a następnie wykonać komendy tam opisane.

Zbudowanie systemu z kodu źródłowego przy pomocy Docker'a:

- Pobranie najnowszej wersji repozytorium FE oraz repozytorium BE
- Zmiana kodu proxy w pliku package.json w repozytorium FE na <http://django:8000>
- Odpalenie skryptu docker\_build\_network.sh
- W obu repozytoriach odpalenie skryptu docker\_build.sh, a następnie docker\_run\_with\_network.sh
- Zarówno na FE jak i BE trzeba ustawić odpowiednie pliki .env, które pozwolą na połączenie się użytkownika z bazami danych Oracle oraz Firebase. Nie podajemy ich informacji globalnie ze względów bezpieczeństwa. W przypadku chęci skorzystania z systemu lokalnie wymagany jest kontakt z administratorami i ustawienie odpowiedniej wartości takich plików.
- Taki proces pozwala już na odpalenie [aplikacji](#)

Konfiguracja systemu:

Aby skonfigurować system trzeba skorzystać z poziomu administratora, który Django proponuje:

- [Globalnie](#)
- [Lokalnie](#)

Do tego celu trzeba mieć specjalnie utworzone konto administratora. Domyślnie istnieje tylko jedno takie konto, choć administratora jest w stanie tworzyć innych administratorów oraz zarządzać ich uprawnieniami, a także konfiguracją systemu.

Dodatkowe uprawnienia administratora

Administrator systemu w panelu administracyjnym Django pod adresem /admin/ może:

1. Tworzyć konta dla innych administratorów
2. Dodawać, edytować, usuwać kategorię prac oraz klasy szkolne

3. Dodawać, edytować oraz usuwać aukcje. W czasie tego procesu wybiera plik np. jpg z pracą ucznia, który stanie się NFT

#### Instrukcja aktualizacji oprogramowania:

W celu aktualizacji oprogramowania należy pobrać najnowszą wersję z repozytorium, po czym odpalić skrypty docker\_build.sh i docker\_run.sh. Warto zwrócić uwagę, że dotychczasowa wersja kodu będzie działała do momentu, w którym zdecydujemy się ją wyłączyć aby odpalić bardziej aktualną wersję. Oznacza to, że czas poświęcony na zastępowanie wersji może zostać skrócony do minimum potrzebnego na zatrzymanie i uruchomienie innego kontenera Dockerowego.

#### Tworzenie kopii zapasowych i odtwarzanie systemu:

W przypadku wystąpienia awarii krytycznych kluczowe jest zapewnienie bezpieczeństwa w warstwie trwałości. Ponieważ korzystamy z dwóch dobrze znanych i szeroko stosowanych baz danych: Oracle oraz FireBase przekazujemy im część odpowiedzialności związanej z backupami baz danych. Każdy dostawca baz danych powinien spełniać restrykcyjne wymogi dotyczące backupów bazy. Jeżeli chodzi o globalny dostęp do aplikacji, aby odpalić ją na innym serwerze wystarczy pobrać kod z repozytorium, ustawić odpowiednie IP i uruchomić Dockera. Spowoduje to upublicznenie aplikacji na innym adresie IP, co z kolei powoduje znaczne zwiększenie bezpieczeństwa aplikacji.

## UX/UI

### Aktorzy

#### Użytkownik

Osoba korzystająca z aplikacji bez dodatkowych uprawnień. Ma możliwość m.in. przeglądać aukcje, brać w nich udział, oglądać swoje NFT oraz doładować portfel.

#### Administrator

Może dodawać aukcje. Przykładem jest pracownik szkoły.

### Persony

#### Robert Kowalski (Aktor: Użytkownik)

Jest młodym chłopcem pełnym pomysłów. Jego ulubionym zajęciem w wolnym czasie jest tworzenie dzieł sztuki i innych kreatywnych form ekspresji siebie. Robert chce podzielić się swoimi pracami z ludźmi i zarobić pieniądze dla swojej szkoły więc aktywnie korzysta z serwisu.

- Robert chce wystawić swoją najnowszą pracę na licytacje w celach promocyjnych
- Robert chciałby porównać swoje umiejętności tworzenia dzieł z rówieśnikami, więc przegląda ich prace
- Robert chciałby pokazać swojemu koledze, że docenia jego sztukę więc aktywnie uczestniczy we wszystkich jego aukcjach

#### Mariusz Kowalski (Aktor: Użytkownik)

Jest tatą Roberta, jego celem jest wspieranie szkoły syna poprzez zakup NFT utworzonych przez uczniów. Mariuszowi bardzo podoba się szkolna inicjatywa, ponieważ sam w wolnym czasie zajmuje się inwestowaniem w NFT i kryptowaluty. Mariusz jest średniozamożnym 35-letnim mężczyzną, więc może być czynnym uczestnikiem aukcji.

- Mariusz chce przeglądać wystawione przez Roberta i jego rówieśników prace
- Mariusz chce kupować NFT syna, pokazując mu że go wspiera
- Mariusz chce pokazywać znajomym zakupione przez niego szkolne NFT

#### Beata Kowalska (Aktor: Użytkownik)

Jest babcią Roberta, chciałaby obserwować działalność wnuka, a może nawet od czasu do czasu brać udział w licytacjach. Nie jest jednak zaawansowanym użytkownikiem komputera, więc ma nadzieję, że strona będzie przyjazna i łatwa w obsłudze.

- Beata chce obserwować działalność wnuka
- Beata chciałaby od czasu do czasu brać udział w licytacjach
- Beata zależy na tym, żeby strona była łatwa w obsłudze

#### Borys (Aktor: Administrator)

Jest nauczycielem plastyki w szkole Roberta. Najbardziej w swojej pracy lubi możliwość obserwacji rozwoju swoich podopiecznych. Serwis NFT umożliwia mu śledzenie postępów swoich uczniów i dostosowywanie metod nauczania.

- Borys chce przeglądać prace swoich uczniów w celu śledzenia ich postępów
- Borys aktywnie wspiera swoich uczniów podbijając stawki ich prac na licytacjach
- Borys dał uczniom szansę na dodatkową ocenę. Mogą ją zdobyć przez uzbieranie największej kwoty na aukcji wśród innych uczniów

## User Stories

### Aktorzy

Lista User Stories:

		Create	Read	Update	Delete
1	AUCTIONS_NFT	Tworzenie aukcji (Administrator)	Wyświetlanie posiadanych NFT, Sortowanie wyświetlonych NFT Wyświetlanie aktualnych/przyszłych aukcji (Użytkownik)	Utworzenie aukcji wraz z wrzuceniem NFT (Administrator)	Usunięcie aukcji (Administrator)
2	USER	Założenie konta w aplikacji (Użytkownik) Założenie konta innemu administratorowi (Administrator)	Wyświetlenie stanu konta (Użytkownik) Wyświetlenie informacji użytkownika (Użytkownik)	Aktualizacja danych użytkownika, Wpłacanie środków do portfela (Użytkownik)	Usunięcie konta w aplikacji (Użytkownik)
3	AUCTIONS_BID	Wzięcie udziału w aukcji (Użytkownik)	Odczytanie aktualnej oferty (Użytkownik)	Wygranie aukcji (Użytkownik)	Wycofanie swojego zakładu (Użytkownik)
4	AUCTIONS_SCHOOL CLASS	Utworzenie klasy (Administrator)	Wyświetlenie NFT przypisanych do klasy (Użytkownik)	Zmiana danych klasy (Administrator)	Usunięcie klasy (Administrator)
5	AUCTIONS_NFTCATEGORY	Utworzenie kategorii NFT (Administrator)	Wyświetlenie NFT przypisanych do kategorii (Użytkownik)	Zaktualizowanie kategorii NFT (Administrator)	Usunięcie kategorii NFT (Administrator)

### Przypadki użycia

### Aktorzy

USE CASE	Wzięcie udziału w aukcji  <a href="https://www.figma.com/file/8IM5ptaarBoVXB2rM2Zd6w/Udzial_w_aukcji?node-id=0%3A1&amp;t=ObtTvOubW1q1Em7X-0">https://www.figma.com/file/8IM5ptaarBoVXB2rM2Zd6w/Udzial_w_aukcji?node-id=0%3A1&amp;t=ObtTvOubW1q1Em7X-0</a>
Description	Użytkownik chciałby wziąć udział w aukcji wybranego NFT
Used by	Chęć wzięcia udziału w aukcji
Preconditions	Aukcja, która go interesuje jest aktualnie w toku
Success end condition	Użytkownik chociaż raz przebił aktualną (na tamten moment) kwotę
Failed end condition	Użytkownik nie posiadał odpowiednich środków i nie udało mu się przebić oponentów
Actors	Użytkownik
Trigger	
DESCRIPTION	
1	Użytkownik wyszukuje / wyświetla mu się na stronie głównej aukcja (figma, ekran 2)
2	Użytkownik wchodzi w aukcję, wyświetla mu się prawidłowo kwota, która się aktualizuje, a także zdjęcie NFT (figma, ekran 3)
3	Kliką przycisk licytuje, wpisuje kwotę, która jest większa niż dotychczasowa (figma, ekran 4)
4	Użytkownik dostaje wiadomość, że jest osobą, która aktualnie prowadzi aukcję (figma, ekran 5)
EXTENTIONS	
2	Kwota aktualizuje się w momencie potwierdzenia zalicytowania innego użytkownika, przebijającego kwotę (+/- czas odpowiedzi serwisu)
3	Serwis sprawdza, czy użytkownik ma wystarczającą ilość pieniędzy

VARIATIONS	
1	Użytkownik wyszukuje aukcję za pomocą filtra, albo widzi aukcję, która go interesuje na stronie głównej
USE CASE	Wyświetlenie posiadanych NFT  <a href="https://www.figma.com/file/HnIVI5hEjeOlEZBSDLWtt/przegladanie_nft?node-id=0%3A1&amp;t=lfBPLd2NRsk6c5WD-0">https://www.figma.com/file/HnIVI5hEjeOlEZBSDLWtt/przegladanie_nft?node-id=0%3A1&amp;t=lfBPLd2NRsk6c5WD-0</a>
Description	Użytkownik chciałby wyświetlić wszystkie posiadane NFT
Used by	Cheć obejrzenia posiadanych NFT
Preconditions	Użytkownik ma dostęp do aplikacji oraz posiada przynajmniej jedno NFT
Success end condition	Użytkownik wszedł na ekran z posiadanymi NFT, a obrazki wczytały się prawidłowo
Failed end condition	Użytkownikowi nie udało się wejść na ekran lub obrazki nie wczytały się prawidłowo
Actors	Użytkownik
Trigger	
DESCRIPTION	
1	Użytkownik loguje się do systemu (figma, ekran 1) <a href="https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A11&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11">https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A11&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11</a>
	Użytkownik na stronie głównej kliką w przycisk NFT's (figma, ekran 2) <a href="https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A5&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11">https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A5&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11</a>
2	Użytkownik jest przekierowywany na następny ekran (figma, ekran 3) <a href="https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A6&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11">https://www.figma.com/proto/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A6&amp;scaling=min-zoom&amp;page-id=1%3A2&amp;starting-point-node-id=1%3A11</a>
3	Użytkownik jest w stanie swobodnie przeglądać wyświetcone w galerii NFT (figma, ekran 4)
EXTENTIONS	
3	Jeżeli pojawia się więcej obrazów, możliwe jest ich scrollowanie
USE CASE	Wyświetlenie aktualnych/przyszłych aukcji
Description	Użytkownik chciałby dowiedzieć się, jakie aukcje są przewidziane w najbliższym czasie
Used by	Cheć przejrzenia aktualnych/przyszłych aukcji
Preconditions	Użytkownik ma dostęp do przeglądania aukcji
Success end condition	Użytkownikowi udało się przejrzeć zaplanowane aukcje
Failed end condition	Użytkownikowi nie udało się przejrzeć zaplanowanych aukcji
Actors	Użytkownik
Trigger	
DESCRIPTION	

1	Użytkownik wchodzi w ekran aukcje
2	Użytkownik widzi listę aukcji, które są zaplanowane na najbliższy czas, posortowaną datami zakończenia aukcji (najpierw aukcje, które zakończą się jako pierwsze)
EXTENTIONS	
2	Aukcje, które wyświetlają się klientowi mogą być do niego dopasowane na podstawie poprzednich, w których brał udział

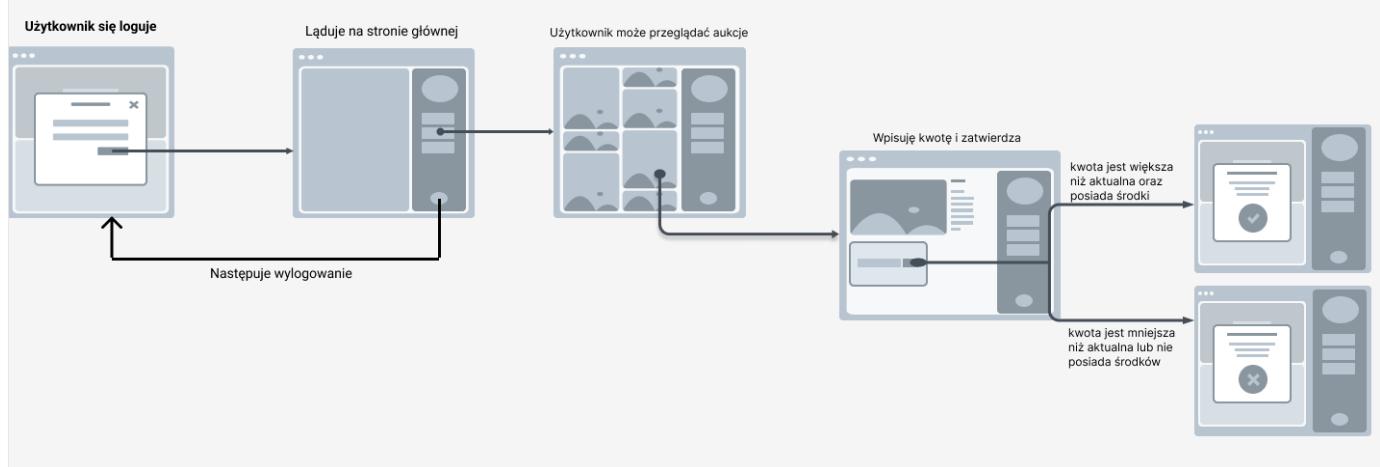
USE CASE	Wpłacenie środków do portfela
Description	Użytkownik chciałby doładować swoje konto, aby był w stanie brać udział w aukcjach
Used by	Chęć doładowania portfela, aby móc brać udział w aukcjach
Preconditions	Użytkownik jest zarejestrowany i ma dostęp do płatności internetowych
Success end condition	Użytkownik wpłacił na portfel wewnętrzny aplikacji taką kwotę, o jaką jego konto zostało pomniejszone
Failed end condition	Użytkownikowi nie udało się doładować portfela, kwota zniknęła z konta a nie powiększyła portfela lub powiększyła portfel, ale nie ubyło z konta
Actors	Użytkownik
Trigger	
DESCRIPTION	
1	Użytkownik wchodzi na ekran doładowania portfela
2	Użytkownik otwiera stronę przeznaczoną do bezpiecznych płatności online
3	Wpisuje kwotę, o którą chciałby powiększyć swój portfel
4	Dokonuje płatności internetowej
5	Po zakończeniu tej płatności jego portfel powinien powiększyć się o kwotę, która zniknęła z konta

USE CASE	Dodanie pracy ucznia do bazy danych
Description	Administrator chciałby, żeby praca ucznia pojawiła się w bazie danych, aby następnie mógł utworzyć aukcję na daną pracę
Used by	Chęć dodania pracy ucznia do bazy danych
Preconditions	Osoba wykonująca tę czynność ma pozycję administratora systemu, a praca ucznia jest zdigitalizowana.
Success end condition	Administratorowi udaje się dodać pracę ucznia do bazy, jest w stanie otworzyć okno do utworzenia aukcji dla danej pracy
Failed end condition	Praca ucznia nie została dodana do bazy lub została dodana nieprawidłowo, a administrator nie może utworzyć na niej aukcji
Actors	Administrator
Trigger	
DESCRIPTION	
1	Administrator otwiera swój panel
2	Klikna przycisk dodaj pracę, który otwiera popup
3	Administrator wybiera pracę ze swojego komputera, która ma zostać załadowana do bazy

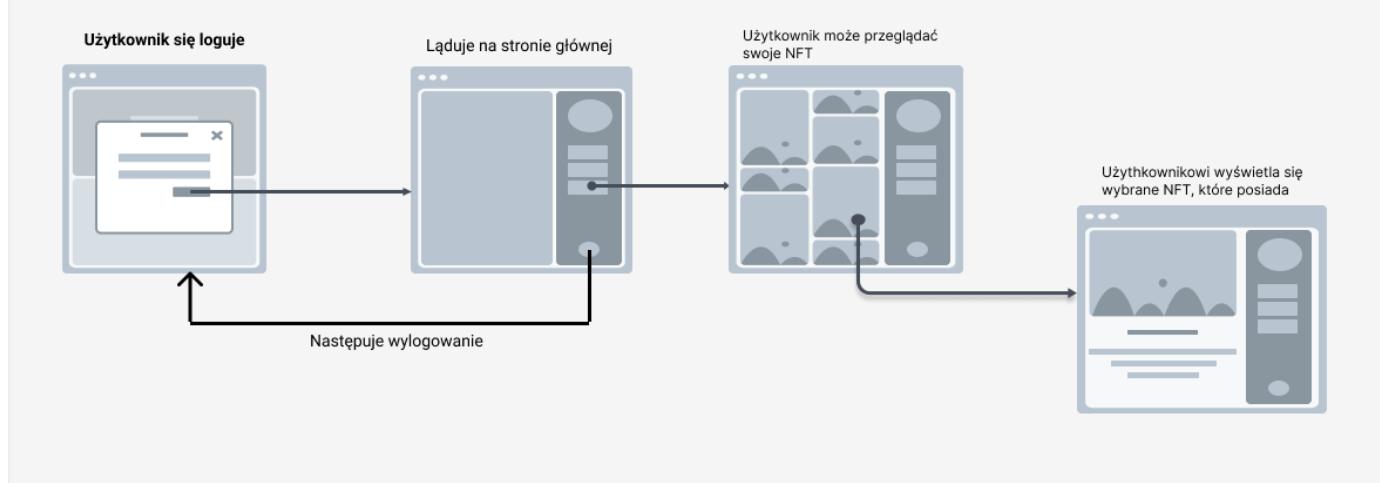
4	Transakcja przebiega pomyślnie
5	Po zakończeniu transakcji, administrator kliknie przycisk utwórz aukcję dla danego NFT i jest w stanie taką aukcję stworzyć

## Wireflow

Branie udziału w aukcji: [https://www.figma.com/file/8IM5ptaarBoVXB2rM2Zd6w/Udzial\\_w\\_aukcji?node-id=0%3A1&t=DMfjvh6o6t4Sav15-1](https://www.figma.com/file/8IM5ptaarBoVXB2rM2Zd6w/Udzial_w_aukcji?node-id=0%3A1&t=DMfjvh6o6t4Sav15-1)



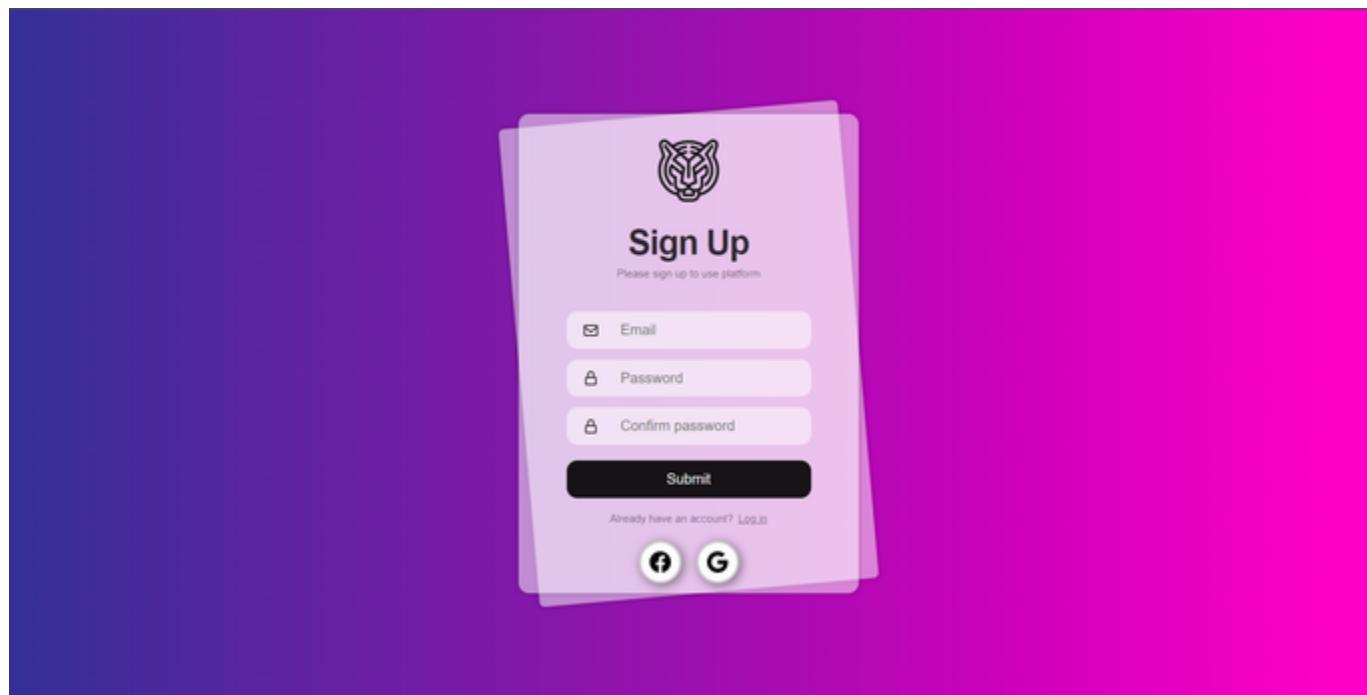
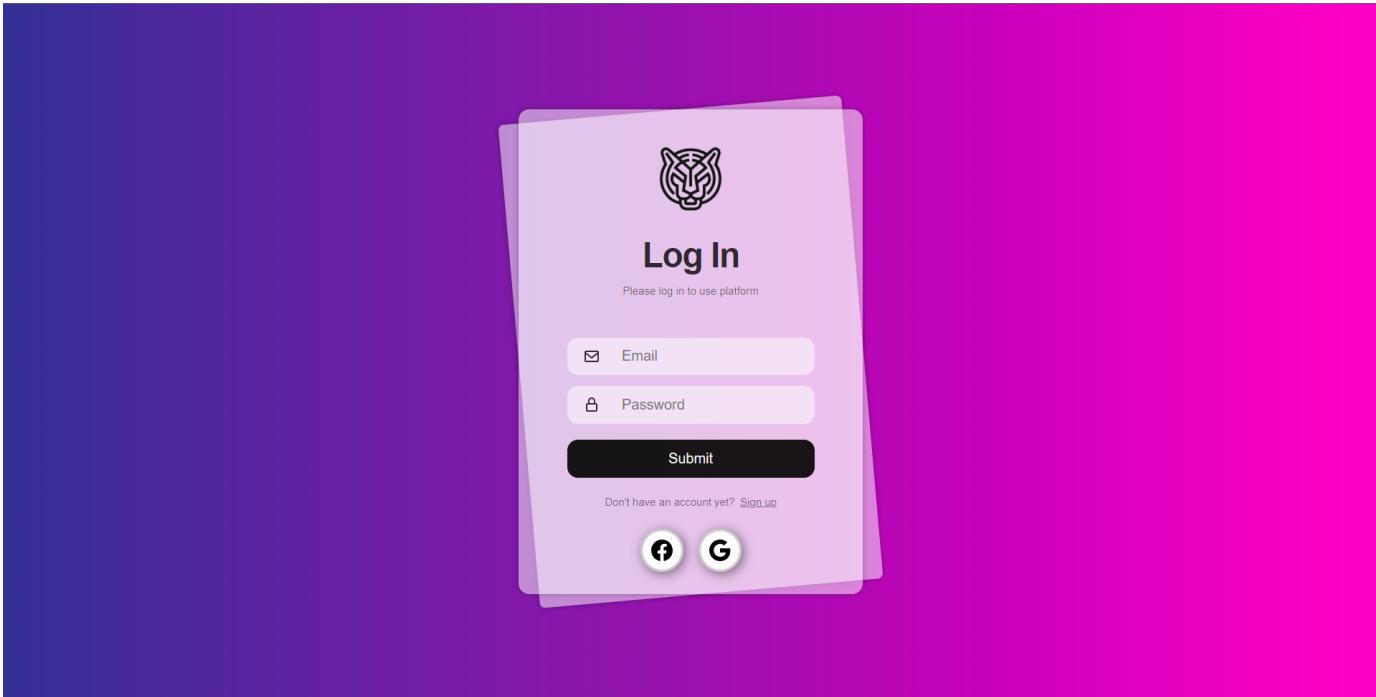
Przeglądanie swoich NFT: [https://www.figma.com/file/HnIVl5hEjeOlxEzWBSDWtt/przegladanie\\_nft?node-id=0%3A1&t=CsxuEfgDncQBre3V-1](https://www.figma.com/file/HnIVl5hEjeOlxEzWBSDWtt/przegladanie_nft?node-id=0%3A1&t=CsxuEfgDncQBre3V-1)

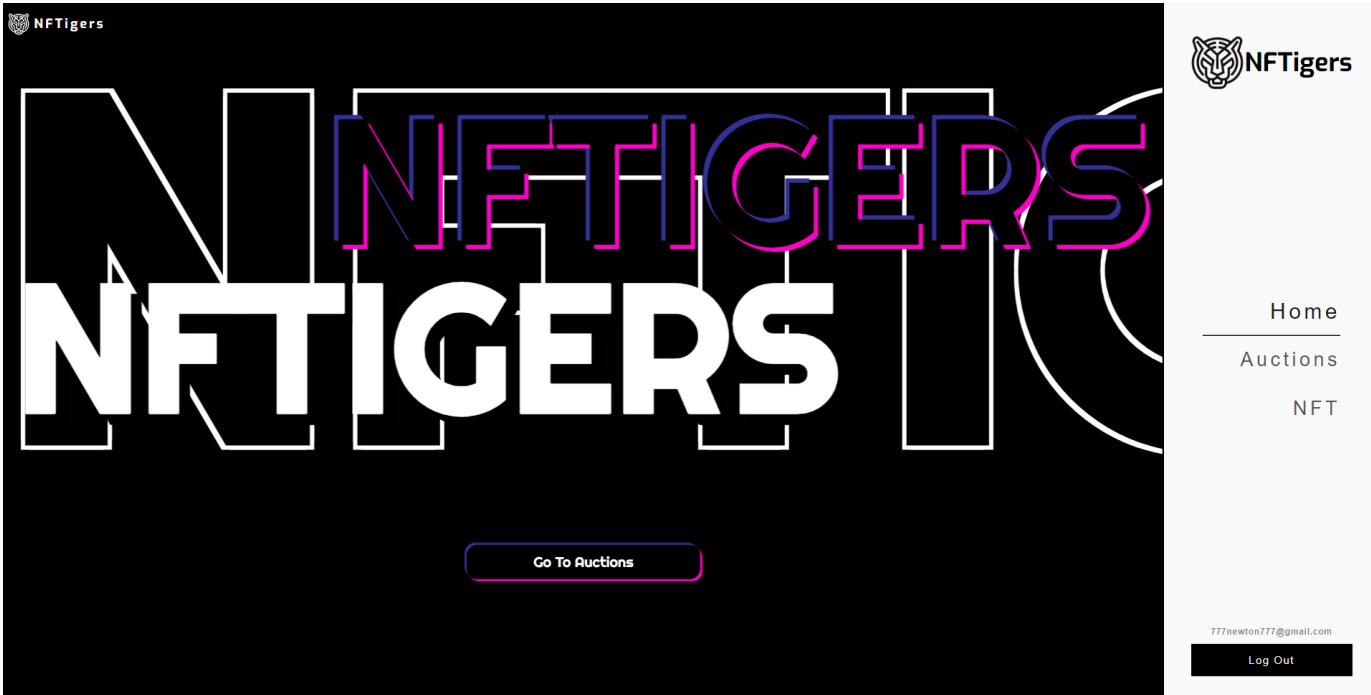


## Ekrany

Desktop 2022.12.20 - 15.10.43.01 (online-video-cutter.com).mp4

<https://www.figma.com/file/h3jKcxsERbYE7YDFufCylb/Untitled?node-id=1%3A2&t=Hy52lyAgHJswUHkW-1>





Domenowy słownik pojęć

#### Domenowy słownik pojęć:

- Użytkownik - Standardowy użytkownik systemu, najczęściej uczeń szkoły lub jego rodzina. Ma ograniczony dostęp do funkcjonalności systemu.
- Administrator - osoba odpowiedzialna za daną szkołę w systemie. Ma możliwość tworzenia klas, uczniów, aukcji, a także może tworzyć NFT z prac ucznia.
- Praca ucznia - Zdigitalizowana stworzona przez ucznia szkoła praca, gotowa, aby przekształcić ją na NFT.
- NFT - Praca ucznia, która została zmintowana. Gotowa do wystawienia na aukcję
- Mintowanie - przekształcanie danych cyfrowych w aktywa cyfrowe zarejestrowane za pomocą technologii blockchain. Dzięki temu procesowi pliki odpowiadające cyfrowym produktom są przechowywane w zdecentralizowanej bazie danych.
- Blockchain - Blockchain jest definiowany jako rejestr zdecentralizowanych danych, które są bezpiecznie współużytkowane.
- Aukcja - forma sprzedaży NFT, w której nabywcą zostaje osoba oferująca najwyższą cenę w określonym przedziale czasowym.

#### Modelowanie Zagrożeń

- Awaria Bazy Danych Oracle
  - Opis
  - Podejście
- Awaria bazy Firebase
  - Opis
  - Podejście
- Awaria Microsoft Azure
  - Opis
  - Podejście
- Kradzież danych wrażliwych użytkowników
  - Opis
  - Podejście
- Kradzież konta
  - Opis
  - Podejście
- Mintowanie NFT
  - Opis
  - Podejście
- Przechwycenie pakietów komunikacji
  - Opis
  - Podejście
- Celowa zmiana parametrów wywołania
  - Opis
  - Podejście
- Zmiana zapytania SQL

- [Opis](#)
- [Podejście](#)
- [Łatwy dostęp do serwera](#)
  - [Opis](#)
  - [Podejście](#)
- [Środki pieniężne użytkownika](#)
  - [Opis](#)
  - [Podejście](#)
- [Inne błędy w działaniu aplikacji](#)
  - [Opis](#)
  - [Podejście](#)

Poniżej znajdują się potencjalne zagrożenia jakie mogą pojawić się w naszym systemie. Dla każdego z nich, oprócz opisu, przedstawiliśmy nasze podejścia, czyli sposób radzenia sobie z zagrożeniem lub powód rezygnacji z zajmowania się danym problemem.

#### **Awaria Bazy Danych Oracle**

##### **Opis**

Awarii w skutek klęski żywiołowej bądź aktu terroru/wandalizmu może ulec baza danych Oracle. Jest ona dostarczona przez nasz wydział (baza ora2). W razie znieszczenia serwerowni mogą zostać utracone dane potrzebne do funkcjonowania naszej aplikacji, w tym dane, do którego użytkownika należy dane NFT. Może się to wiązać ze stratami majątkowymi użytkowników, gdyż NFT zostały zakupione z wykorzystaniem prawdziwych pieniędzy

##### **Podejście**

Należałoby zrobić backup bazy danych w miejscu oddalonym od fizycznej lokalizacji pierwszej bazy danych. Ryzyko jednokrotnej awarii w obu miejscach jest znacznie mniejsze.

Z racji na to, że jest to tylko projekt studencki oceniany przez nasz wydział. Jeśli zostanie uszkodzona serwerownia bazy danych, najpewniej będzie to sytuacja krytyczna dla całego wydziału lub całej uczelni i zajęcia przestaną odbywać się w sposób normalny. Potencjalne zagrożenie z racji na wspomniany fakt pomijamy.

#### **Awaria bazy Firebase**

##### **Opis**

Dane do logowania naszych użytkowników są przechowywane w bazie Firebase od firmy Google. W razie awarii moglibyśmy stracić dane wrażliwe naszych użytkowników, a także możliwość zalogowania się użytkowników do naszego systemu. Wiązać by się to mogło ze stratami majątkowymi użytkowników, którzy zostaliby pozbawieni bez dostępu do swoich dóbr materialnych jakimi są NFT.

##### **Podejście**

Zgodnie z prawem, wszyscy dostawcy baz danych muszą sami zadbać o bezpieczeństwo oferowanych usług, w tym za przechowywanie odpowiednich backupów. W razie awarii i braku takiego zabezpieczenia, odpowiedzialność za szkody ponosić będzie firma Google. Ma ona na tyle wysoką renomę i poziom usług, że sytuacja ta jest znikomo prawdopodobna.

#### **Awaria Microsoft Azure**

##### **Opis**

Serwer naszej aplikacji postawiony jest na maszynie wirtualnej na platformie Azure firmy Microsoft. W razie awarii platformy Azure (np. serwerów) żaden użytkownik nie będzie w stanie połączyć się do naszego serwera i korzystać z naszej aplikacji.

##### **Podejście**

Duża część internetu stoi na Microsoft Azure, dlatego w razie awarii zapewne nikt niewróci uwagi na to, że nie może wejść do naszej aplikacji. Ponadto, za platformę odpowiedzialną jest firma z wysoką renomą, generującą na niej ogromne zyski, dlatego klient (my) powinien czuć się bezpieczny korzystając z niej. W razie wystąpienia zagrożenia odpowiedzialność poniesie firma Microsoft. Choć uważamy, że zagrożenie jest znikomo prawdopodobne.

#### **Kradzież danych wrażliwych użytkowników**

##### **Opis**

Jak zostało to wcześniej wspomniane, dane potrzebne do logowania się użytkowników są zarządzane przez usługę Firebase. My przechowujemy jedynie identyfikator oraz nazwę użytkownika. Nie są to dane wrażliwe.

## **Podejście**

To częste zagrożenie u nas nie występuje. Za zabezpieczenie danych odpowiada dostawca Firebase, czyli Google.

### **Kradzież konta**

#### **Opis**

W naszym systemie użytkownicy obracają swoimi pieniędzmi. W związku z tym, nieuwierzytelny użytkownik nie może mieć dostępu do wykorzystania środków bądź sprzedaży NFT jednego z naszych klientów. W przypadku, gdy w jakiś sposób uzyska login i hasło jednego z naszych klientów, jest w stanie zalogować się na konto i zrobić z jego własnością co chce.

## **Podejście**

Za bezpieczeństwo swoich haseł i loginów są odpowiedzialni użytkownicy. Hasła powinny być silne i niepowtarzane w innych systemach. Żaden serwis nie jest w stanie zapewnić 100% ochrony. Potencjalnym dodatkowym zabezpieczeniem mogłoby być dwufazowe logowanie (np. przy użyciu Google Authenticator). W związku jednak z ograniczeniem czasowym na projekt, nie wprowadzamy tego rozwiązania w naszym systemie i przenosimy pełną odpowiedzialność na użytkownika.

### **Mintowanie NFT**

#### **Opis**

Potencjalnym zagrożeniem aplikacji może być próba wpłynięcia na kod podczas procesu mintowania NFT.

## **Podejście**

Ponieważ nasze rozwiązanie jest tylko projektem studenckim, a nie rozwiązaniem, które będzie stosowane na szeroką skalę stwierdziliśmy, że wykorzystywanie serwisów zewnętrznych do tworzenia blockchainu jest zbyt skomplikowane. W związku z tym napisaliśmy kod, który symuluje tworzenie blockchainu. Nasz kod korzysta z funkcji hashującej sha 256. Nowy hash tworzony jest na podstawie starego hasha oraz pewnej informacji mówiącej, kto aktualnie jest posiadaczem NFT. Takie podejście pozwala prześledzić historię, a także zapewnia integralność danych, a osoba posiadająca NFT w naszej aplikacji jest jego wyłącznym posiadaczem.

### **Przechwytcie pakietów komunikacji**

#### **Opis**

Częstym sposobem ataku jest wykorzystanie słabości protokołów komunikacyjnych. Ze względu na charakter komunikacji w ogólnodostępnej sieci Internet opartej na protokole IP, nie można zakładać bezpieczeństwa przesyłanych danych. Cała komunikacja może zostać podsłuchana, przechwycona lub sfałszowana.

## **Podejście**

W związku z potencjalnym zagrożeniem, szyfrujemy transmisję protokołu za pomocą TLS i wykorzystujemy protokół HTTPS zamiast HTTP.

### **Celowa zmiana parametrów wywołania**

#### **Opis**

W rzeczywistości każde naciśnięcie przycisku na stronie powoduje wysłanie do serwera żądania strony i listy parametrów. Parametry te mogą być widoczne dla użytkownika w oknie przeglądarki (metoda GET) lub nie (metoda POST). W związku z tym użytkownik może w prosty sposób wysłać specjalnie spreparowane żądanie, podmieniając na przykład wartość niektórych parametrów lub dodając nowe. Zapytanie takie w przypadku nieprawidłowo zaprojektowanej aplikacji mogłoby dać mu dostęp do danych, do których nie powinien mieć uprawnień.

## **Podejście**

W żądaniu przesyłany będzie Guid użytkownika a po stronie serwera nastąpi sprawdzenie, czy użytkownik o danym identyfikatorze ma dostęp do zasobu, do którego się odwołuje.

### **Zmiana zapytania SQL**

#### **Opis**

Poważnym i często spotykanym błędem jest wklejenie parametru, bez sprawdzania jego przesłanej wartości, bezpośrednio w tekst, który jest z kolei wysyłany do bazy jako treść zapytania. Przy odpowiednio przygotowanym ciągu znaków użytkownik może zupełnie zmienić treść zapytania i spowodować na przykład uszkodzenie bazy danych. Zagrożenie jest powszechnie znane jako SQLInjection.

## Podejście

W naszej aplikacji mamy wyraźnie oddzieloną warstwę logiki od warstwy dostępu do danych, w związku z czym, treść zapytania nie jest wklejana bezpośrednio. Zapytania są już gotowe i wystawione przez pewne klasy. Aplikacja na podstawie parametrów z metod HTTP jedynie okraja zapytania. Ponadto parametry metod HTTP będą silnie typowane. Choć język Python jest oparty na typowaniu dynamicznym, to przy wykorzystaniu odpowiednich mechanizmów, jesteśmy w stanie wykryć, czy użytkownik nie przesłał specjalnie spreparowanego zapytania np. w miejsce w którym oczekujemy zmiennej całkowitoliczbowej.

## Łatwy dostęp do serwera

## Opis

Maszyna wirtualna w chmurze stanowi bardzo częsty cel hakerów, ze względu na mnogość danych na niej się znajdujących - w tym danych, które umożliwiają zarządzanie działaniem całej maszyny wirtualnej oraz wszystkich znajdujących się na niej aplikacji. Częstotliwość występowania takich ataków jest przeważająca - sprawdziliśmy to własnoręcznie na innej maszynie wirtualnej, do której logowanie odbywało się za pomocą hasła. Podejrzaliśmy pliki znajdujące się w katalogu `/var/log/`, które pokazywały próby zalogowania na maszynę. Przez pierwsze 100 sekund działania serwisu aż 83 razy próbowało dokonać logowania za pomocą metody brute-force. To pokazuje, że takie ataki są na porządku dziennym i trzeba się przed nimi bronić.

## Podejście

W celu łączenia się z naszą maszyną wirtualną korzystamy z połączenia ssh przy wykorzystaniu klucza.

## Środki pieniężne użytkownika

## Opis

Nasz system to system aukcyjny, więc ważną kwestią jest zadbanie o bezpieczeństwo środków użytkownika. Bezpieczne wpłacanie, wypłacanie, przechowywanie informacji. Wszelkie kwestie związane ze środkami powinny zostać odpowiednio zabezpieczone.

## Podejście

W związku z tym, że jest to tylko projekt studencki, a połączenie z serwisami płatniczymi wymaga sporej ilości czasu, w naszym systemie środki użytkownika nie są rzeczywistymi pieniędzmi. Jest to tylko symulacja, więc nie widzimy zagrożenia, które mogłyby płynąć z tak przyjętego podejścia.

## Inne błędy w działaniu aplikacji

## Opis

W fazie implementacji i testowania istnieje ryzyko niewykrycia wszystkich błędów aplikacji.

## Podejście

Jeśli starczy nam czasu, zaimplementujemy system logowania błędów, dzięki któremu, będziemy w stanie na bieżąco monitorować pojawiające się bugi. Możliwymi rozwiązaniami są: logowanie do pliku tekstowego znajdującego się na maszynie wirtualnej lub logowanie do Azure Application Insight.