# Systemy mobilne i multimedia - Raport JurassicDex

# Kasper Radom, Łukasz Welka 27 maja 2025

# 1 Wstęp

Celem projektu **JurassicDex** było stworzenie aplikacji multimedialnej prezentującej dinozaury w atrakcyjnej, edukacyjnej formie. Aplikacja została podzielona na trzy główne kategorie: dinozaury lądowe, wodne oraz latające, z czego każda zawiera cztery unikalne gatunki wraz z opisami, grafikami, filmami oraz materiałami audio. Dodatkowo aplikacja zawiera moduł quizu, pozwalający użytkownikowi sprawdzić swoją wiedzę.

# 2 Opis zawartości aplikacji

Aplikacja JurassicDex zawiera 12 dinozaurów podzielonych na trzy kategorie:

# Dinozaury lądowe

- Brachiozaur
- Stegozaur
- Triceratops
- Tyrannosaurus rex

## Dinozaury wodne

- Liopleurodon
- Ichtiozaur
- Mozazaur
- Plejozaur

## Dinozaury latające

- Dimorphodon
- Rhamphorhynchus
- Quetzalcoatlus

#### Pteranodon

Dla każdego z dinozaurów przygotowano szczegółowy opis zawierający: nazwę, okres występowania, rozmiary, sposób odżywiania, ciekawostki, ilustracje, materiał wideo oraz audio.

W aplikacji dostępny **Quiz**, który pozwala użytkownikowi sprawdzić swoją wiedzę o dinozaurach. Quiz składa się z 5 losowo wybranych pytań z większej puli, a każde pytanie zawiera 7 możliwych odpowiedzi, z których tylko jedna jest poprawna. Po udzieleniu odpowiedzi użytkownik od razu dowiaduje się czy była ona poprawna. Dzięki temu aplikacja pełni również funkcję edukacyjną, użytkownik od razu poznaje prawidłową odpowiedź i może wyciągnąć wnioski. Aby przejść do kolejnego pytania, użytkownik musi przesunąć palcem od prawej do lewej na ekranie.

Aplikacja zawiera również zakładkę **O nas**, która zawiera podstawowe informacje o autorach projektu. Można tam znaleźć imiona i nazwiska twórców, krótkie bio oraz zdjęcia. Sekcja ta ma na celu zwiększenie autentyczności projektu oraz umożliwia kontakt w przypadku pytań lub sugestii dotyczących aplikacji.

## 3 Materialy promocyjne

W ramach projektu wykonano następujące elementy promocyjne:

• Logo aplikacji – zaprojektowane z wykorzystaniem programu graficznego Inkscape.



Do stworzenia figur użyto narzędzia do tworzenia wielokątów ( $\mathbf{B}$ ), a do dalszej edycji i zaokrąglania krawędzi zastosowano narzędzie węzłów ( $\mathbf{N}$ ). Kolorem zastosowanym w projekcie loga był kolor w zapisie heksadecymalnym:  $\mathbf{c68841ff}$ , co nadaje logo ciepły, bursztynowy odcień, nawiązujący do klimatu prehistorycznego i barw kojarzonych z epoką dinozaurów.

Ulotka informacyjna – zawiera skrócony opis funkcjonalności aplikacji oraz grafikę promującą JurassicDex. Została przygotowana przy użyciu narzędzia Adobe Express, na bazie gotowego szablonu z zakładki Flyer, a następnie odpowiednio zmodyfikowana, aby lepiej oddawała klimat i stylistykę wykorzystaną w aplikacji.



Interaktywny przewodnik po dinozaurach z opisami, ilustracjami i quizem!

• Sylwetki projektantów – krótki opis zespołu odpowiedzialnego za projekt, wraz z fotografią dostępny w aplikacji w zakładce O nas.





JurassicDex

Cześć! Miło Cię widzieć w Jurassicdex - naszej mobilnej encyklopedii dinozaurów. To aplikacja, którą stworzyliśmy z pasji do technologii, nauki i – oczywiście prehistorycznych gadów.

Jesteśmy studentami informatyki stosowanej na Politechnice Wrocławskiej:

- · Łukasz Welka wcześniej studiował matematykę, obecnie rozwija się w kierunku informatyki. Interesuje się logiką, algorytmiką i nowoczesnymi technologiami, które pomagają łączyć naukę z codziennością
- · Kasper Radom absolwent informatyki technicznej, obecnie student na informatyce stosowanej. Lubi tworzyć aplikacje, które są nie tylko funkcjonalne, ale też przyjazne





JurassicDex

Mamy nadzieję, że nasza aplikacja przypadnie Ci do gustu i pozwoli odkrywać świat dinozaurów w nowoczesny sposób

#### Dziękujemy, że tu jesteś!

jurassicdex@hotmail.com















## 4 Pliki multimedialne

W projekcie **JurassicDex** wykorzystano różnorodne pliki multimedialne, które wspierają interaktywny i atrakcyjny charakter aplikacji.

#### Formaty plików:

- Ilustracje (.jpg) wykorzystywane głównie do prezentacji dinozaurów. Pliki zostały odpowiednio przeskalowane i zoptymalizowane pod kątem wydajności aplikacji.
- Materiały wideo (.mp4) krótkie filmy przedstawiające animacje dinozaurów.
- Materiały audio (.mp3) efekty dźwiękowe prezentujące odgłosy dinozaurów.
- Gify (.gif) animacja wykorzystywana na stronie startowej.
- Dodatkowe grafiki (.png) grafiki pomocnicze wykorzystywane w aplikacji i materiałach promocyjnych.

#### Prawa autorskie:

- Obrazy dinozaurów pochodzą głównie ze strony Pixabay, która udostępnia zasoby graficzne na licencji umożliwiającej darmowe, niekomercyjne wykorzystanie bez konieczności przypisywania autorstwa.
- Grafiki domeny publicznej część ilustracji została pozyskana z zasobów Wikimedia Commons, gdzie zaznaczone zostały jako objęte domeną publiczną (public domain) i tym samym wolne od ograniczeń licencyjnych.
- Wideo materiały wideo zostały stworzone z wykorzystaniem narzędzia AI RunwayML, które umożliwia generowanie animacji na podstawie wprowadzonego tekstu i obrazów. Filmy wygenerowane w ten sposób mogą być wykorzystywane w projektach edukacyjnych zgodnie z warunkami licencji serwisu.
- Audio materiały dźwiękowe pochodzą ze strony Freesound, z której zostały wykorzystane materiały należące do domeny publicznej(CC0).
- Informacje o dinozaurach teksty opisujące dinozaury zostały opracowane na podstawie ogólnodostępnych źródeł edukacyjnych i popularnonaukowych (np. Wikipedia), a następnie sparafrazowane i przystosowane do formy aplikacji, z zachowaniem oryginalnego sensu i faktów. Żadne treści nie zostały skopiowane bezpośrednio.
- Materiały własne logo aplikacji, grafiki promocyjne, gify oraz zrzuty ekranu zostały wykonane samodzielnie przez autorów projektu.

## 5 Kroje pisma i efekty tekstowe

W projekcie zastosowano domyślną rodzinę krojów pisma systemowego (**FontFamily.Default**), zapewniającą spójność wizualną i poprawne wyświetlanie na różnych urządzeniach mobilnych.

Dla różnych elementów interfejsu użytkownika użyto odmiennych stylów tekstu:

- titleLarge wykorzystywany do nagłówków i tytułów sekcji. Zastosowano pogrubienie (FontWeight.Bold), rozmiar czcionki 22 sp, interlinię 24 sp oraz rozstaw znaków 0.5 sp, co zapewnia dobrą czytelność i hierarchię informacji.
- bodyLarge używany do tekstu głównego i opisów. Zastosowano standardową wagę pisma (FontWeight.Normal), rozmiar 16 sp, interlinię 24 sp i rozstaw znaków 0.5 sp. Styl ten gwarantuje wygodę w czytaniu dłuższych treści na ekranach urządzeń mobilnych.

Dzięki zachowaniu jednolitej typografii z subtelnym rozróżnieniem pomiędzy nagłówkami a treścią, aplikacja zapewnia przejrzysty i przyjazny dla użytkownika interfejs.

# 6 Wymagania funkcjonalne i niefunkcjonalne

#### Funkcjonalne:

- Możliwość przeglądania kategorii i dinozaurów,
- Odtwarzanie materiałów multimedialnych (grafiki, wideo, dźwięki),
- Uruchamianie quizu i uzyskiwanie wyniku,
- Możliwość sprawdzenia poprawności odpowiedzi przed przejściem do kolejnego pytania w quizie,
- Dostęp do zakładki "O nas" z informacjami o autorach aplikacji.

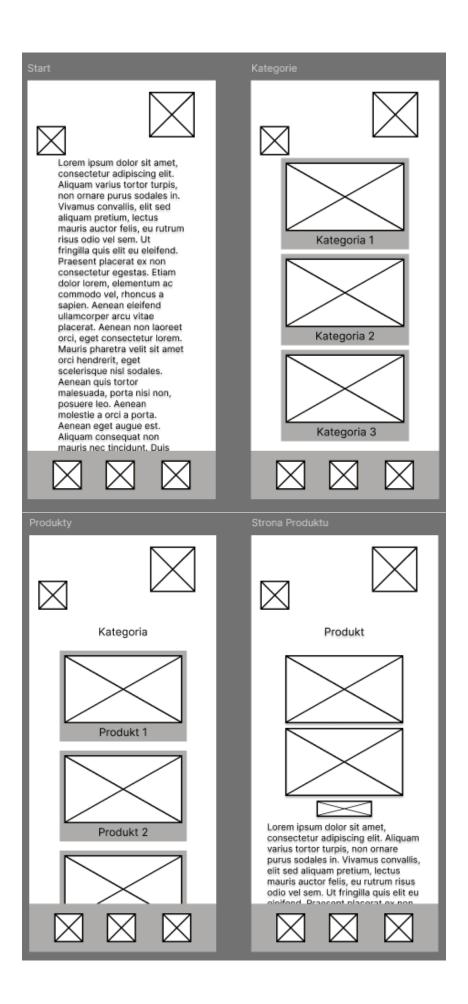
#### Niefunkcjonalne:

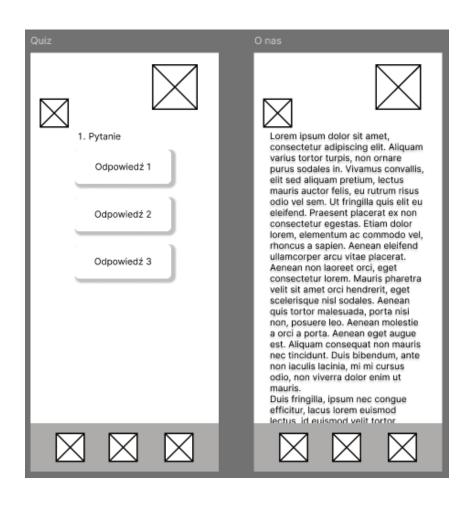
- Intuicyjny interfejs,
- Szybkość działania aplikacji,
- Kompatybilność z urządzeniami mobilnymi,
- Poprawne działanie odtwarzania audio i wideo,
- Optymalizacja rozmiaru plików multimedialnych dla płynnego działania aplikacji.

## 7 Schemat i makieta projektu

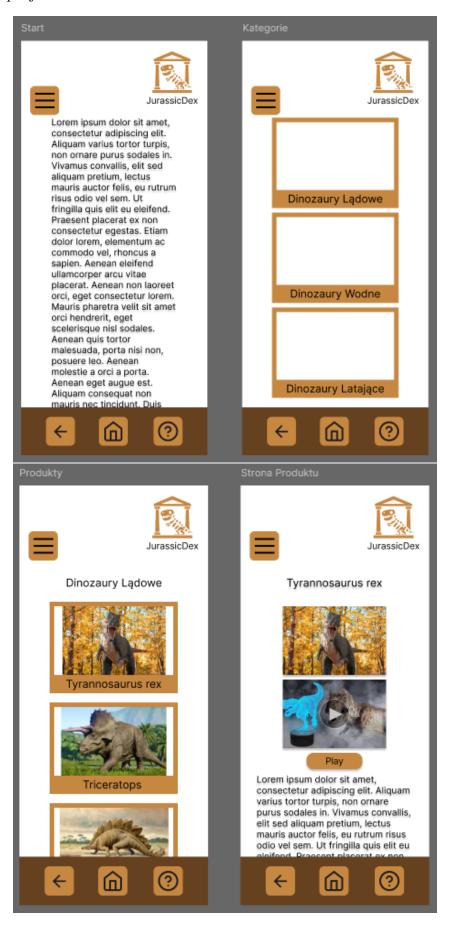
Schemat oraz makieta projektu zostały stworzone z wykorzystaniem narzędzia Figma, które umożliwia efektywne projektowanie interfejsów użytkownika oraz wizualizację struktury aplikacji.

Schemat projektu:





## Makieta projektu:





# 8 Implementacji aplikacji

Aplikacja **JurassicDex** została napisana w języku Kotlin z użyciem Jetpack Compose — nowoczesnego narzędzia do budowy interfejsu użytkownika w Androidzie. W celu zapewnienia przejrzystości, testowalności oraz łatwego zarządzania stanem aplikacji, przyjęto architekturę MVVM (Model-View-ViewModel).

#### 8.1 Model

Warstwa Model odpowiada za logikę aplikacji i przechowywanie danych. W projekcie obejmuje ona następujące klasy:

- CategoryData klasa odpowiedzialna za przechowywanie informacji o pojedynczym elemencie kategorii. Składa się z:
  - name nazwy dinozaura, która wyświetlana jest jako podpis pod grafiką,
  - imageRes identyfikatora zasobu graficznego R.drawable, odpowiadającego ilustracji danego dinozaura,
  - destination ścieżki nawigacyjnej w aplikacji, do której użytkownik zostanie przeniesiony po kliknięciu.
- CategoriesData klasa grupująca wiele obiektów typu CategoryData, służąca do zorganizowania widoku kategorii (dinozaury lądowe, wodne, latające). Składa się z:

- name tytułu kategorii, np. Dinozaury Lądowe,
- categories listy obiektów CategoryData, reprezentujących pojedyncze karty do kliknięcia,
- showTitle wartości logicznej określającej, czy tytuł kategorii ma być widoczny.
- DinosaurData klasa danych przechowująca kompletne informacje do wyświetlenia na stronie danego dinozaura. Składa się z:
  - title tytułu strony, zawierającego nazwę dinozaura,
  - images listy identyfikatorów zasobów graficznych R. drawable przedstawiających danego dinozaura,
  - text1 sformatowanego, wypunktowanego tekstu AnnotatedString z najważniejszymi informacjami na temat dinozaura, zawierającego np. pogrubienia, kolory lub inne cechy stylizacji,
  - text2 tekstu pomocniczego, np. ciekawostek lub dodatkowych informacji o dinozaurze,
  - video opcjonalnego identyfikatora zasobu wideo R.raw, jeśli dany dinozaur ma przypisany materiał wideo,
  - audio opcjonalnego identyfikatora zasobu dźwiękowego R.raw, np. dźwięku wydawanego przez dinozaura.
- QuestionData klasa reprezentująca pojedyncze pytanie w quizie. Składa się z:
  - question treści pytania wyświetlanego użytkownikowi,
  - answers listy możliwych odpowiedzi (ciągi znaków), spośród których użytkownik może wybrać jedną,
  - correctAnswer poprawnej odpowiedzi, służącej do weryfikacji odpowiedzi użytkownika.

#### 8.2 ViewModel

Warstwa ViewModel pośredniczy pomiędzy warstwą widoku a modelem, zarządza logiką interfejsu użytkownika i stanem aplikacji. W projekcie obejmuje ona następujące klasy:

- QuizViewModel odpowiada za logikę działania quizu. Przechowuje listę pytań jako listę obiektów QuestionData.
- DinosaurViewModel zarządza danymi szczegółowymi wyświetlanymi dla wybranego dinozaura. Udostępnia tytuł, teksty, obrazy, dźwięk i wideo związane z danym okazem. Zawiera również funkcję getDinosaurByName, która na wyszukuje informacje o dinozaurze na podstawie jego nazwy. Przyjmuje jako argument nazwę dinozaura (title) i przeszukuje listę dinosaurs. Zwraca pierwszy obiekt typu DinosaurData, którego pole title zgadza się z przekazaną nazwą. Jeżeli żaden pasujący element nie zostanie znaleziony, funkcja zwraca null.

• CategoryViewModel – odpowiada za udostępnianie danych kategorii Categories-Data dostępnych w aplikacji, takich jak dinozaury lądowe, latające czy wodne. Szczególnym przypadkiem jest obiekt listy, zawierający zbiór reszty kategorii. Odpowiada on za ekran z wyborem kategorii dinozaura. Klasa zawiera równię metodę getCategoryByName pozwalającą na wyszukiwanie konkretnej grupy kategorii na podstawie jej nazwy. Zwraca pierwszy pasujący obiekt CategoriesData lub null, jeśli nie znaleziono dopasowania.

#### 8.3 View

Warstwa View odpowiada za prezentację danych użytkownikowi oraz reagowanie na jego interakcje.

Widoki w projekcie korzystają z danych udostępnianych przez klasę ViewModel. Komponenty takie jak przyciski, obrazy czy teksty są dynamicznie generowane na podstawie danych dostarczonych z warstwy Model. Każda sekcja aplikacji (np. quiz, lista kategorii, szczegóły dinozaurów) posiada osobny ekran, który korzysta z odpowiedniego ViewModela i danych źródłowych.

Wszystkie zmiany stanu (np. rozpoczęcie odtwarzania dźwięku, wybór odpowiedzi w quizie) są obsługiwane za pomocą obserwacji stanu oraz reaktywnego odświeżania widoków, zgodnie z paradygmatem deklaratywnego UI. Użytkownik ma dostęp do przyjaznego, interaktywnego i responsywnego interfejsu.

Projekt skłąda się z plików odpowiedzialnych za generowanie kolejnych ekranów:

- **Start.kt** ekran startowy aplikacji, zawiera tytuł, animację oraz krótki tekst zachęcający do korzystania z aplikacji. Stanowi punkt wejścia dla użytkownika.
- CategoryScreen.kt ekran wyświetlający listę kategorii dinozaurów (lądowe, wodne, latające) oraz zbioru wyżej wymienionych. Kategorie są prezentowane w formie klikalnych paneli, które przekierowują do szczegółowych podkategorii lub listy dinozaurów.
- DinosaurScreen.kt ekran szczegółów danego dinozaura. Wyświetla nazwę, ilustracje, opisy oraz multimedia (np. dźwięk, wideo), korzystając z danych typu DinosaurData.
- QuizScreen.kt ekran quizu edukacyjnego dla użytkownika. Zawiera pytania, możliwe odpowiedzi i sprawdza poprawność zaznaczenia. Dane pobierane są z klasy QuizViewModel. Po przejściu quizu aplikacja pokazuje użytkownikowi ilość zdobytych punktów i daje mu możliwość ponownego przejścia quizu. Pytania dobierane są losowo z puli zadeklarowanej w warstwie ViewModel. Kolejność wyświetlania odpowiedzi jest również losowa.
- OnasScreen.kt ekran zawierający informacje o twórcach aplikacji. Ma charakter informacyjny.

## 8.4 Reszta plików

Poza plikami wymienionymi jako elementy architektury MVVM, warto również wspomnieć o kilku innych niezbędnych do poprawnego działania aplikacji:

• **MeinActivity.kt** - główna klasa Activity w aplikacji, odpowiadająca za inicjalizację interfejsu użytkownika (UI) i zarządzanie nawigacją między ekranami.

W metodzie onCreate realizowane są istotne zadania inicjalizacyjne. W pierwszej kolejności aktywowany jest tryb edge-to-edge pozwalający na pełne wykorzystanie powierzchni ekranu. Następnie następuje ukrycie systemowych pasków (nawigacyjnego i statusu), które mogą być tymczasowo wyświetlane poprzez gest przesunięcia palcem, co zapewnia bardziej immersyjne wrażenia użytkownika.

System nawigacji oparty jest na komponencie Animated Nav<br/>Host, który umożliwia płynne przejścia między ekranami. Zdefiniowano pięć głównych tras nawigacyjnych: ekran startowy <a href="startowy start">startowy start</a>, ekran kategorii <a href="category/cathegoryTypeName">categoryTypeName</a>, ekran szczegółów dinozaura <a href="dinozaurName">dinozaurName</a>, ekran informacyjny onas oraz ekran quizu <a href="quiz">quiz</a>.

Każde przejście między ekranami wzbogacone jest animacją składającą się z efektu przesunięcia w poziomie (slideInHorizontally) oraz stopniowego pojawiania się (fadeIn), co nadaje aplikacji płynny i profesjonalny charakter. Ekrany dynamiczne, takie jak ekran kategorii czy szczegółów dinozaura, przyjmują odpowiednie parametry przekazywane przez system nawigacji.

• **Type.kt** - definiuje style typograficzne używane w aplikacji. Obiekt Typography zawiera zestaw predefiniowanych stylów tekstowych, które zapewniają spójność wygladu tekstu w całej aplikacji.

Główny obiekt *Typography* składa się z dwóch podstawowych stylów tekstowych. Pierwszy z nich, *bodyLarge*, przeznaczony jest dla standardowego tekstu w aplikacji i charakteryzuje się normalną grubością czcionki (*FontWeight.Normal*), rozmiarem 16.sp oraz odstępem między wierszami wynoszącym 24.sp, co zapewnia czytelność dłuższych fragmentów tekstu. Dodatkowo zastosowano minimalne odstępy między literami (0.5.sp) dla lepszej prezentacji.

Drugi styl, *titleLarge*, służy do wyświetlania większych nagłówków i wyróżnionych tekstów. Wyróżnia się pogrubioną czcionką (*FontWeight.Bold*) i większym rozmiarem (22.sp), przy zachowaniu tej samej wysokości linii co *bodyLarge* (24.sp) dla zachowania spójności wizualnej. Podobnie jak w przypadku stylu podstawowego, zastosowano niewielki odstęp między literami (0.5.sp), który poprawia czytelność większych napisów.

Oba style wykorzystują domyślną rodzinę czcionek systemowych (FontFamily.Default), co zapewnia naturalne integrowanie się z preferencjami użytkownika i spójność z systemem operacyjnym. Zdefiniowane style typograficzne mogą być łatwo rozszerzane o dodatkowe warianty tekstowe w miarę rozwoju aplikacji, zachowując przy tym spójny system projektowy oparty na zasadach Material Design 3.

• Theme.kt - definiuje niestandardowy motyw aplikacji JurassicDex wykorzystujący Material Design 3 w Jetpack Compose. Główna funkcja *JurassicdexTheme* automatycznie dostosowuje wygląd aplikacji do preferencji użytkownika, obsługując tryb ciemny i dynamiczne kolory na Androidzie 12+.

Motyw oferuje dwa schematy kolorystyczne - ciemny z brązowymi odcieniami (*MediumBrown*, *DarkBrown*) i jasny z jaśniejszymi brązami (*LightBrown*, *Brown*). Funkcja inteligentnie wybiera schemat: na nowszych urządzeniach z włączonymi dyna-

micznymi kolorami wykorzystuje systemowe palety dopasowane do tapety użytkownika, a na starszych - statyczne zestawy kolorów w zależności od aktywnego trybu.

Motyw integruje również wcześniej zdefiniowane style tekstowe (*Typography*), zapewniając spójność typografii w całej aplikacji. Dzięki bazowaniu na *MaterialTheme*, wszystkie komponenty automatycznie dostosowują swój wygląd do aktualnego schematu, co ułatwia utrzymanie jednolitego interfejsu. To rozwiązanie pozwala aplikacji oferować spersonalizowane doświadczenia wizualne, jednocześnie zachowując zgodność z zasadami Material Design i preferencjami użytkownika.

- Color.kt definiuje paletę kolorów dla aplikacji, zawierającą zestaw spójnych odcieni brązu i kolory tła dostosowane do trybów jasnego i ciemnego. W trybie jasnym dominują ciepłe odcienie brązu główny kolor LightBrown (C68841) i akcentujący Brown (654120), z czarnymi ikonami (IconsColorLight) na białym tle (BackgroundLight). Wersja ciemna wykorzystuje głębsze brązy MediumBrown (A27020) i DarkBrown (492A0E), z białymi ikonami (IconsColorDark) na jasnoszarym tle (BackgroundDark EBEBEB). Ta starannie dobrana paleta kolorystyczna zapewnia spójność wizualną aplikacji, jednocześnie dbając o odpowiedni kontrast i czytelność w obu trybach wyświetlania.
- SharedComponents.kt zawiera trzy główne komponenty interfejsu użytkownika dla aplikacji:
  - Nagłówek (Header) wyświetla logo aplikacji i tytuł "JurassicDex", które po kliknięciu przenoszą użytkownika na ekran startowy.
  - Menu hamburger (HamburgerMenuOverlay) to rozwijane menu z ikoną w lewym górnym rogu, zawierające opcje nawigacyjne do sekcji: Start, Kategorie, Quiz i O nas. Każda opcja po kliknięciu odpowiednio przekierowuje użytkownika, z animowanym opóźnieniem dla płynniejszego przejścia.
  - Stopka (Footer) znajduje się na dole ekranu i zawiera trzy przyciski nawigacyjne: powrót, stronę główną i quiz, wszystkie w kolorystyce dopasowanej do motywu aplikacji. Komponenty są responsywne, uwzględniając orientację ekranu. Każdy element interaktywny posiada tagi testowe ułatwiające automatyzację testów.

# 9 Testy

## 9.1 Testy automatyczne

Aplikacja JurassicDex przeszła kompleksowe testy, łączące metody manualne i zautomaty-zowane z wykorzystaniem narzędzi Jetpack Compose do weryfikacji interfejsu. Głównym założeniem procesu testowego było potwierdzenie stabilności działania, zgodności z założeniami projektowymi interfejsu użytkownika oraz poprawnego funkcjonowania wszystkich elementów nawigacyjnych. Testy obejmowały różne scenariusze użytkowania, konfiguracje urządzeń i stany aplikacji, co pozwoliło na wszechstronną weryfikację jakości oprogramowania.

Testy zostały podzielone na kilka głównych kategorii, sprawdzających różne aspekty funkcjonalności aplikacji:

#### • Testy podstawowej nawigacji:

- Weryfikacja poprawnego wyświetlania ekranu startowego
- Przejścia między ekranami (Kategorie, Quiz, O nas)
- Nawigacja do konkretnych sekcji (dinozaury lądowe, wybrany dinozaur)

#### • Testy menu hamburger:

- Sprawdzenie poprawności wyświetlania i ukrywania rozwijanego menu
- Weryfikacja działania poszczególnych opcji menu

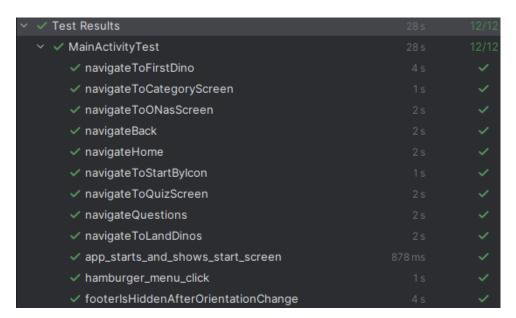
#### • Testy stopki aplikacji:

- Działanie przycisku powrotu
- Funkcjonalność przycisku home
- Przejście do quizu ze stopki
- Zachowanie stopki przy zmianie orientacji ekranu

#### • Testy dodatkowych funkcji:

- Nawigacja przez kliknięcie logo
- Zachowanie aplikacji przy zmianie orientacji urządzenia

Pozytywne wyniki testów wskazują na poprawną implementacje wyżej wymienionych funkcjonalności.



## 9.2 Testy manualne

Przeprowadzone zostały szczegółowe testy manualne na różnych urządzeniach (smartfonach i tabletach), aby sprawdzić, czy interfejs pozostaje spójny i funkcjonalny niezależnie od rozmiaru ekranu. Weryfikowaliśmy m.in.:

• Responsywność UI – czy elementy (przyciski, menu) zachowują poprawne proporcje i układ na różnych rozdzielczościach,

- Poprawność nawigacji czy menu (w tym rozwijane hamburger menu) działa płynnie, a przejścia między ekranami są intuicyjne,
- Wyświetlanie multimediów czy filmy i dźwięki odtwarzają się bez problemów, a zdjęcia dinozaurów prawidłowo się powiększają po kliknięciu.
- **Spójność motywu** czy tryb jasny/ciemny działa poprawnie na wszystkich widokach,
- Obsługa gestów czy aplikacja reaguje na przewijanie, dotknięcia i inne interakcje zgodnie z oczekiwaniami,
- Mechanizm punktacji czy system poprawnie zlicza punkty za prawidłowe odpowiedzi oraz czy nie przyznaje punktów za błędne wybory. Testowane były różne kombinacje odpowiedzi, w tym częściowo poprawne i całkowicie błędne,
- Walidacja poprawnych odpowiedzi czy każde pytanie ma jednoznacznie określoną prawidłową odpowiedź i czy system prawidłowo je rozpoznaje. W tym celu wyniki quizu porównywane były z wiedzą ekspercką na temat dinozaurów,
- Zachowanie aplikacji po zakończeniu quizu czy po rozwiązaniu wszystkich pytań użytkownik otrzymuje podsumowanie z właściwą liczbą punktów oraz czy możliwe jest ponowne rozpoczęcie quizu bez błędów,
- **Spójność wizualna** czy interfejs quizu (przyciski, licznik punktów, podświetlenie odpowiedzi) działa poprawnie na różnych urządzeniach i rozdzielczościach

Dodatkowo sprawdzone zostało, czy aplikacja zachowuje stabilność podczas zmiany orientacji ekranu oraz czy nie występują opóźnienia w ładowaniu treści. Wszystkie testowane funkcje działały prawidłowo, a ewentualne błędy wizualne (np. przesunięte elementy) były na bieżąco poprawiane.

# 10 Zasady UI/UX

Projekt aplikacji **JurassicDex** opiera się na prostych, ale konsekwentnych zasadach projektowania interfejsu użytkownika (UI) i doświadczenia użytkownika (UX). Główne założenia to: czytelność, spójność wizualna, intuicyjność oraz estetyka nawiązująca do tematyki prehistorycznej.

## 10.1 Kolorystyka

Zastosowano dwa tryby kolorystyczne: **jasny** i **ciemny**, które automatycznie dopasowują się do ustawień systemowych użytkownika. Paleta barw została dobrana w taki sposób, aby oddać naturalny, "ziemisty" klimat dinozaurów i środowiska epoki mezozoicznej, mając na uwadze kontrast kolorów, zgodnie z wytycznymi **WCAG** na poziomie **AA**.

#### • Tryb jasny:

- Tło: #FFFFF,

- Kolor tła ikon: #C68841,

- Kolor tła paska nawigacji: #654120,

- Ikony i teksty: #000000.

#### • Tryb ciemny:

- Tło: #EBEBEB,

- Kolor tła ikon: #A27020,

Kolor tła paska nawigacji: #492A0E,

- Teksty: #000000,

- Ikony: #FFFFF.

## 10.2 Zasady projektowe UX

- Czytelność i hierarchia informacji wykorzystanie zróżnicowanej typografii ułatwia szybkie przyswajanie informacji.
- Intuicyjna nawigacja interfejs został podzielony na logiczne sekcje: przegląd gatunków, quiz, zakładka "O nas", co pozwala użytkownikowi łatwo odnaleźć interesujące go treści.
- Spójność graficzna zastosowanie stałej palety kolorów i powtarzalnych wzorców interfejsu (np. przycisków, kart) wzmacnia komfort użytkowania.
- Dostosowanie do urządzeń mobilnych projekt przygotowany został w oparciu o responsywny layout, który działa poprawnie zarówno w pionowej, jak i poziomej orientacji ekranu.
- Efekty interakcji zastosowanie animacji, efektów kliknięcia oraz natychmiastowej informacji zwrotnej (np. w quizie) zwiększa zaangażowanie użytkownika.