

Projekt wykorzystuje poniższe technologie:

- Node.js + Express
- PUG
- React
- MySQL
- Auth0

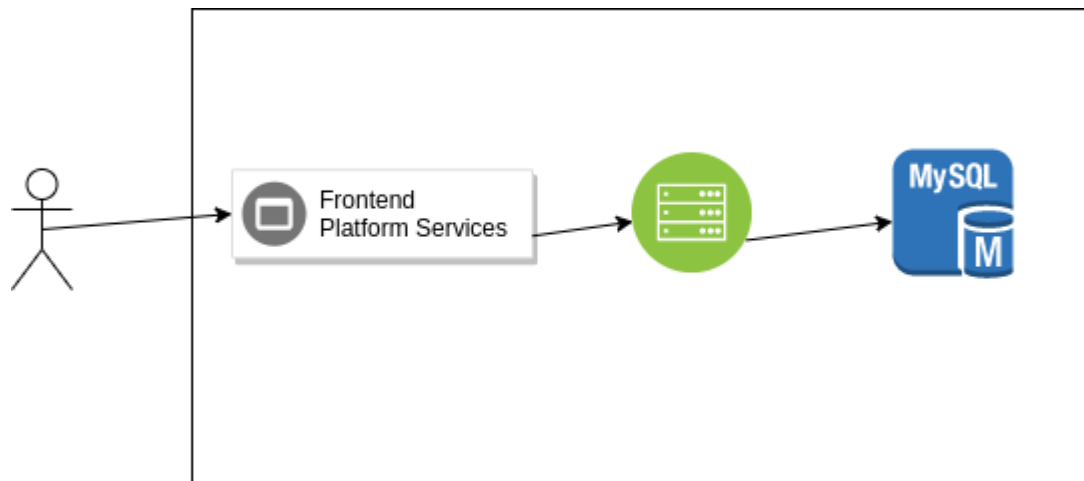
Identyfikacja zagadnienia biznesowego

Celem projektu było stworzenie aplikacji służącej do kolekcjonowania zasobów ulubionych płyt muzycznych. Odbiorcą aplikacji może być każda osoba, które chce zapisać przeprowadzić spis wszystkich swoich ulubionych płyt muzycznych. Funkcjonalność założonej aplikacji może stanowić część większego projektu związanego z czynnością słuchania muzyki jak odtwarzanie utworów, platforma oceny i komentarzy utwor muzycznych.

Wymagania systemowe i funkcjonalne

Aplikacją na mieć możliwość przechowywać listę zespołów, albumu, utworów muzycznych, okładek albumów, posiadać odnośniki do platform muzycznych.

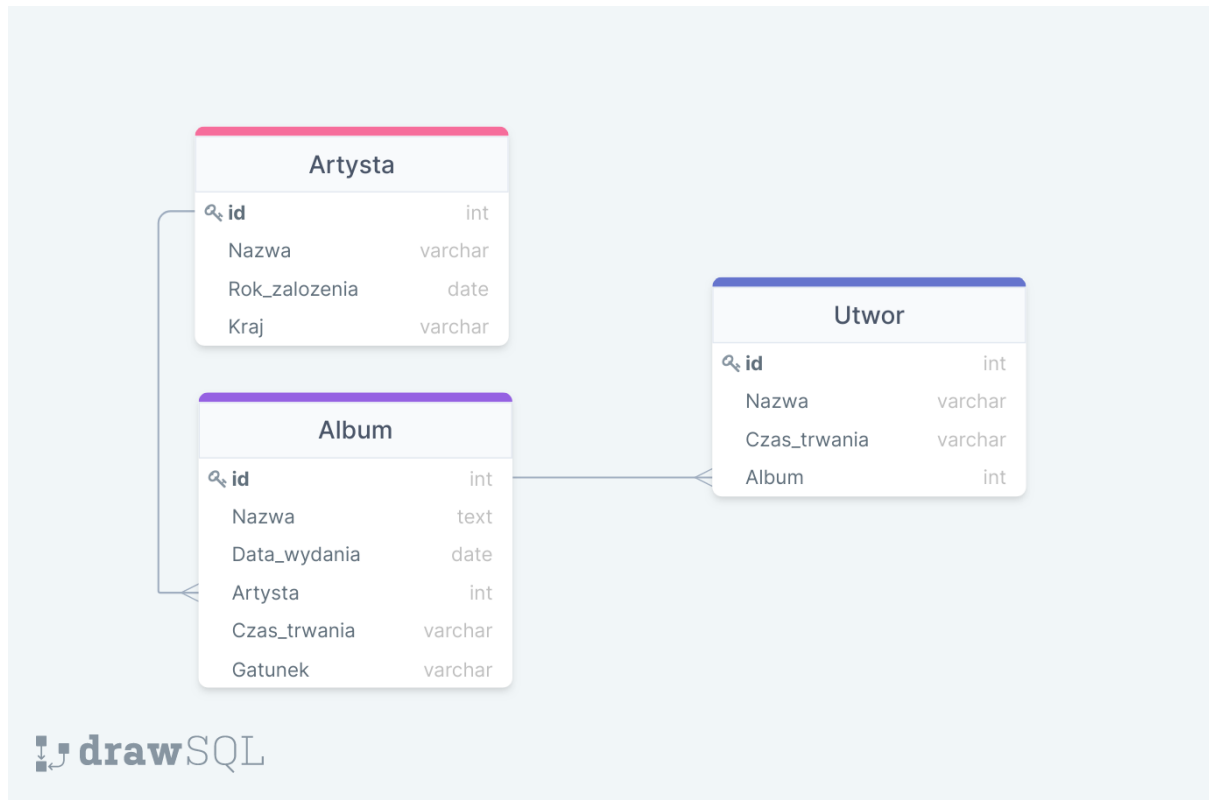
Program po stronie użytkownika powinien działać na wszystkich współczesnych przeglądarkach internetowych, do swojego działania wymagane jest połączenie do serwera. Do uruchomienia aplikacji po stronie serwera wymagany wsparcie na uruchomienia technologii: Node.js + Express, React oraz relacyjnej bazy danych MySQL.



Analiza zagadnienia i jego modelowanie

Aplikacja podzielona jest na frontend, backend oraz bazę danych.

Schemat bazy danych:



Implementacja

W pliku `baza.sql` przygotowano schemat bazy danych dla MariaDB.

Za backend aplikacji odpowiadają `node.js(14)` + `express 4.16.4`.

Backend za pomocą moduły `mysql` łączy się z bazą danych.

Kluczową funkcjonalnością jest funkcja routera `express` poprzez którą zostało stworzone api umożliwiające odczytywanie i aktualizowanie zasób bazy danych aplikacji.

Przykładowo endpoint umożliwiający pobranie listy zespołów:

```
router.get('/api/bands', function(req, res, next) {  
  var zespol = 'SELECT * FROM Artysta ORDER BY Nazwa ASC';
```

```
db.query(zespol, function(error,dane){
  res.json(dane);
});
});
```

Są to funkcje:

- Pobrania zespołów
- Pobrania albumów
- Pobrania utworów
- Pobrania okładki albumy oraz linki do spotify
- Dodania zespołu
- Dodania albumu wraz z utworami
- Usunięcia albumu
- Usunięcia zespołu

Dostęp do strony zarządzania bazą danych aplikacji zabezpieczony odbywa się po uwierzytelnieniu się użytkownika poprzez serwis OAuth oraz aplikacja Auth0. W tym celu użyty został moduł passport.

Pierwotnie aplikacja posiada front end generowany poprzez backend za pomocą PUG. Posiada on pełnię zakładanych funkcjonalności. Ja jego potrzeby utworzone na express routery. Generujące widoki. Aplikacja uruchomić można poprzez adres: <http://localhost:3000>

Następnie wykorzystano React to utworzenia frontendu aplikacji dostępnego pod adresem: <http://localhost:9000>, nie posiada on jednak pełnej funkcjonalności.

Na jego potrzeby zostały utworzone osobne endpoint w kontekście /api.

- Pobranie zespołów oraz analogicznie albumów

```
router.get('/api/bands', function(req, res, next) {
  var zespol = 'SELECT * FROM Artysta ORDER BY Nazwa ASC';
  db.query(zespol, function(error,dane){
    res.json(dane);
  });
});
```

- pobranie szczegółów albumów oraz utworów

```
router.get('/api/albums/tracks/:a', async function(req, res, next) {
  var trackQuery = 'SELECT Utwor.*, Album.Nazwa AS Album FROM Utwor JOIN Album
ON Album.ID=Utwor.Album WHERE Album.ID="' + req.params.a + '"';

  try {
    const [trackResp] = await Promise.all([
      runQuery(trackQuery)
    ]);
```

```

    res.json(trackResp);
  } catch (error) {
    console.log(error);
    res.json(error);
  }
});

```

Do pobrania okładek i linków wykorzystano moduł `spotify-web-api-node`. Który wyszukuje i zwraca pierwszy wynik z api spotify dla danej nazwy albumu

```

router.get('/api/albums/spotify/:a', async function(req, res, next) {
  var albumssQuery = 'SELECT * FROM Album WHERE ID="' + req.params.a + '"

  try {
    const [albumssResp] = await Promise.all([
      runQuery(albumssQuery)
    ]);

    const spotifyRes = await spotifyApi.searchAlbums(albumssResp[0].Nazwa);
    //console.log("Search artists by "Love", spotifyRes.body.albums.items[0]);
    const albumCoverUrl = spotifyRes?.body?.albums?.items?.[0].images?.[1].url;
    const albumLink = spotifyRes?.body?.albums?.items?.[0].external_urls.spotify;

    res.json({albumCoverUrl: albumCoverUrl, albumLink: albumLink});
  } catch (error) {
    console.log(error);
    res.json(error);
  }
});

```

Podsumowanie

Aplikacja spełnia swoje podstawowe funkcjonalności przechowywania albumów wraz z podstawowymi danymi.

Trudności okazało się zaprojektowanie frontendu z pomocą Reacta.

Aplikacja może być dalej rozwijana poprzez następujące funkcjonalności:

- Pobieranie większej ilości informacji i albumach z internetu oraz przechowywanie ich w bazie danych
- Tworzenie spersonalizowanych list dla zalogowanych użytkowników
- Oceny i komentarze pod albumami i zespołami.