

Gaana.com clone

Server

- database structure

```
> mysql -u root -p
```

```
create database gaana;
use gaana;

create table user (id integer primary key auto_increment, firstName
varchar(100), lastName varchar(100), email varchar(100), password
varchar(100), isActive int(1) default 1, created_timestamp TIMESTAMP
default CURRENT_TIMESTAMP);

create table album (id integer primary key auto_increment, title
varchar(100), artistId integer, thumbnail varchar(100), duration
varchar(10), created_timestamp TIMESTAMP default CURRENT_TIMESTAMP);

create table song (id integer primary key auto_increment, title
varchar(100), albumId integer, artistId integer, songFile varchar(100),
duration varchar(10), created_timestamp TIMESTAMP default
CURRENT_TIMESTAMP);

create table artist (id integer primary key auto_increment, firstName
varchar(100), lastName varchar(100), thumbnail varchar(100),
create_timestamp TIMESTAMP default CURRENT_TIMESTAMP);
```

- express application

```
> mkdir server
> cd server
> npm install express mysql2 crypto-js multer cors
```

Client

- portal
 - create portal app

```
> npx create-react-app portal
```

- install required packages

```
> cd portal  
> npm install react-router react-router-dom axios
```

- port: 3002

- requirements

- home
 - list of artists
 - list of albums
- album details
- user signin
- user signup

- **admin panel**

- create portal app

```
> npx create-react-app admin-panel
```

- install required packages

```
> cd portal  
> npm install react-router react-router-dom axios
```

- requirements

- artist
 - add
 - get list
 - edit
 - delete
- album
 - add
 - get list of albums
 - get album info
 - edit
 - delete
- users
 - get list
 - activate / deactivate the user(s)

- port: 3001