**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA………..**

**ĐẶNG XUÂN DUY KHƯƠNG  
TRƯƠNG THANH NAM**

**KHÓA LUẬN TỐT NGHIỆP**

**ỨNG DỤNG WATERMARKING AUDIO VÀO PHÂN PHỐI NHẠC SỐ   
<Tên khóa luận Tiếng Anh>**

**KỸ SƯ/ CỬ NHÂN NGÀNH <TÊN NGÀNH>**

**TP. HỒ CHÍ MINH, 2020**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA………….**

**<TÊN SINH VIÊN> – <MÃ SINH VIÊN>**

**KHÓA LUẬN TỐT NGHIỆP**

**<TÊN KHÓA LUẬN>**

**<Tên khóa luận Tiếng Anh>**

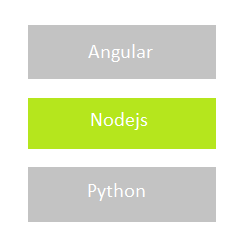
**KỸ SƯ/ CỬ NHÂNNGÀNH <TÊN NGÀNH>**

**GIẢNG VIÊN HƯỚNG DẪN**

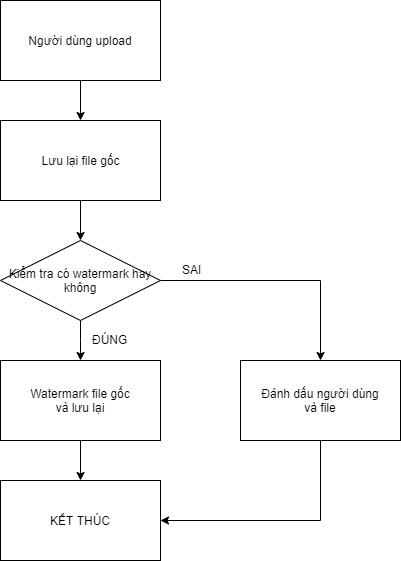
**<TÊN GIẢNG VIÊN HƯỚNG DẪN>**

**TP. HỒ CHÍ MINH, <NĂM>**

1. **MỤC ĐÍCH**  
     
   Ở thời đại giải trí đang phát triển mạnh, các dịch vụ giải trí online như stream phim, ảnh, video đang ngày càng phát triển. Việc có thể nghe nhạc trực tiếp online đang là một phần tất yếu. Các dịch vụ stream nhạc cũng từ đó phát triển như spotify, soundcloud, applemusic, zing mp3, nhaccuatui,…  
     
   Tuy nhiên những nền tảng này lại gặp nhiều khó khăn trong việc quản lý và nhận diện những sản phẩm bị reupload. Làm ảnh hưởng đến doanh thu người phát hành.  
     
   Công nghệ watermarking là một trong những công nghệ có tiềm năng trong việc quản lý các quyền dữ liệu số như phim, ảnh, nhạc,…
2. **TỔNG QUAN**  
   Đồ án sau nhắm đến việc thiết kế và tích hợp 1 hệ thống nhằm ứng dụng công nghệ watermarking vào việc phát hành sản phẩm âm nhạc trên hệ thống số. Đồ án nhằm chứng minh được khả năng tích hợp của công nghệ watermarking vào ứng dụng phát hành sản phẩm âm nhạc, mức độ khả thi, độ an toàn và độ đáng tin cậy.  
     
   Trong phần này chúng ta sẽ bàn về ứng dụng theo hướng nhìn kỹ thuật  
     
   **2.1) TỔNG QUAN CẤU TRÚC HỆ THỐNG**  
   Hệ thống sẽ có 3 thành phần chính: Angular cho front-end, Nodejs cho Web-API và Python cho watermarking và detect-watermarking



Hệ thống dựa vào nền tảng MEANSTACK ( MongoDB, Express, Angular, Nodejs) nhằm tăng tốc độ phát triển và độ linh hoạt của hệ thống.



**2.2) TỔNG QUAN CẤU TRÚC DỮ LIỆU**

**3) TỔNG QUAN THUẬT TOÁN**

Việc ẩn dữ liệu bên trong tín hiệu âm thanh đặc biệt khó khăn, bởi vì con người có khả năng nhận biết âm thanh trên một quãng rất rộng. Ngoài ra khả năng nhận biết những âm thanh nhiễu cũng rất chính xác. Tuy nhiên cũng có vài “lỗ hổng” trong khả năng này. Ví dụ những âm thanh nhỏ thường sẽ bị những âm thanh lớn che khuất. Ngoài ra còn có những âm thanh nền trong môi trường, vì thế đa số trường hợp sẽ bị bỏ qua bởi người nghe.

Để phát triển một hệ thống phát phối nhạc số, việc watermark phải không làm thay đổi chất lượng âm thanh quá lớn, khả năng phát hiện rất thấp, khả năng chống tấn công cao…

Trong các phương pháp ẩn thông tin trong âm thanh, phương pháp ẩn âm thanh trong tiếng vọng (Echo Data Hiding) trở nên phù hợp nhất trong việc bảo vệ tác quyền âm nhạc số.

**ĐIỂM LỢI**

* Khó nhận ra, hoặc sự xuất hiện của tiếng vọng làm âm thanh trở nên hay hơn
* Khó bị thay đổi, khả năng chống tấn công tốt

**ĐIỂM BẤT LỢI**

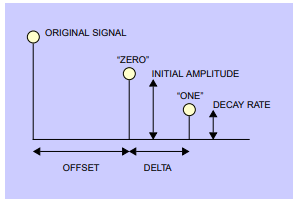
* Tỉ lệ nhúng thấp

Tuy nhiên, khi xét về nhu cầu trong hệ thống. Chúng ta chỉ cần lưu một lượng dữ liệu nhỏ để xác nhận tác quyền của tác giả/nhà phân phối, chứ không cần một file lớn. Vì thế phương pháp EDH được chọn để tích hợp vào hệ thống phân phối nhạc số

**3.1) CƠ SỞ LÝ THUYẾT**

**ECHO DATA HIDING**

Thuật toán Echo Data Hiding rất được chú trọng trong công cuộc nghiên cứu watermarking. Tiếng vọng sẽ được thêm vào tín hiệu âm thanh để chèn dữ liệu vào bản nhạc. Dữ liệu được ẩn đi bằng việc thay đổi ba giá trị chính:



Cho o0, o1, α lần lượt là offset cho bit 0, bit 1 và amptitute. Sau đó kernels tiếng vọng có thể được tính bằng phương trình 1 với δ [n] là hàm Kronecker delta như phương trình 2 để thể hiện cho xung của tín hiệu rời rạc.

(1)

(2)

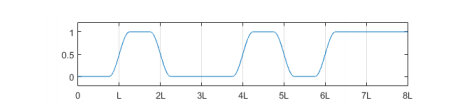
Tính hiệu tiếng vọng sau đó sẽ được tạo ra bằng cách tích chập tính hiệu âm thanh gốc và kernels tiếng vọng.

**ENCODING**

Bản âm gốc sẽ được chia thành nhiều phân khúc bằng với số bits sẽ được gắn vào. Sau đó mỗi phân khúc sẽ được thêm tiếng vọng với offset ứng với dữ liệu bit được gắn vào. Gọi N là số lượng bit cần gắn. L là độ dài của phân khúc. L được chọn sao cho N.L nhỏ hơn độ dài của tín hiệu âm thanh.

Tạo một tính hiệu trộn sẽ giúp tăng tính thuận lợi trong quá trình mã hóa, giúp giảm sự nhiễu âm trong quá trình gắn các phân khúc.

Các bước mã hóa dữ liệu vào đoạn âm thanh như sau:

* Tạo một chuỗi trong đó chứa chữ số không ở đầu.
* Tích chập bản nhạc gốc với để được bản tiếng vọng của tín hiệu âm thanh ứng với
* Tạo một tín hiệu trộn sử dụng những bits được mã hóa vào như hình:  
  
* Thêm tiếng vọng vào tín hiệu âm thanh gốc bằng tín hiệu trộn trong phương trình

**DECODING**

Quá trình giải mã sử dụng việc phân tích tần số. Audio ẩn sẽ được chia thành những phân khúc bằng với số bit cùng với độ dài phân khúc đã được mã hóa. Sau đó lấy phần số thực của tần âm tại điểm có offset trùng với bit tương ứng như phương trình

Với:

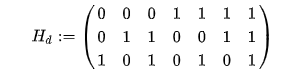
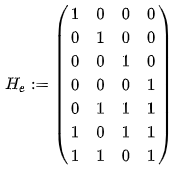
**HAMMING CODE**

Hamming code là một tập các code tự sửa lỗi có thể dùng để phát hiện và tự sửa lỗi có thể xảy ra trong quá trình truyền và lưu dữ liệu từ người gửi đến người nhận. Kỹ thuật này được phát triển **bởi R.W.Hamming.**

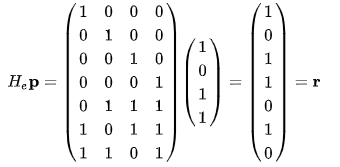
Trong quá trình mã hóa/ giải mã, có nhiều tác nhân gây sai lệch trong tầng số của tiếng vọng, trong đó có thể là lỗi trong việc lưu tập tin, hoặc tấn công cả cố ý và vô ý. Vì vậy việc sửa lỗi bits sẽ tăng độ chính xác trong việc giải mã thông tin.

Ta sẽ sử dụng Haming(7,4) để mã hóa và giải mã đoạn bit truyền vào  
  
**MA TRẬN HAMMING**

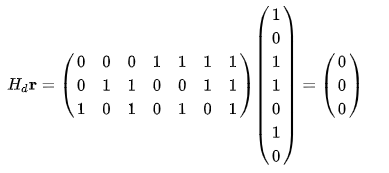
Chúng ta có 2 ma trận HAMMING đi liền với nhau:



Chúng ta truyền một nhóm 4 bits dữ liệu : (1,0,1,1)   
Ta tính tích của và p



Sau đó mã hóa vào đoạn thông tin vào bản nhạc.

Khi giải mã, để kiểm tra có lỗi xảy ra hay không chúng ta sẽ nhân và r

Nếu có lỗi xảy ra kết quả sẽ trả về vị trí lỗi của nó, ví dụ chúng ta sai ở bit thứ 2:



Kết quả trả về tương đương “010” là 2

**3.2) THỰC NGHIỆM THUẬT TOÁN**

**THỰC NGHIỆM**

So sánh giữa 32 mẫu nhạc gồm nhiều thể loại khác nhau Pop, Vpop, Rock, EDM,… Gồm nhiều độ dài khác nhau và sampled khác nhau.  
Sử dụng các thông số:

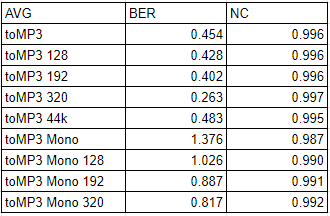
* L = Segment Bits Length = 8192 bit
* O = Offset = 0.04 ms
* D = Delta = 0.02 ms
* Amp = Echo Amplitude = 0.3

Sau đó sử dụng các kiểu tấn công nhằm thay đổi chất lượng như chuyển đổi định dạng, resampling, chuyển sang mono…  
Những file bị tấn công sau đó được giải mã và tính toán các thông số

* BER (Bit Error Rate)
* NC (Normalized Correlation)
* SNR (Sample-to-Noise Ration)

Để có thể đánh giá độ tin cậy của thuật toán.

**KẾT QUẢ**





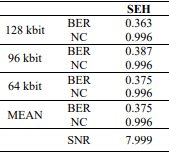
Kết quả cho thấy thuật toán hoàn toàn có thể chống các kiểu tấn công thay đổi định dạng rất tốt, với tỉ lệ BER thấp, hệ thống có thể đảm bảo tính mạnh mẽ trong việc bảo vệ tác quyền âm nhạc.

Với SNR cao trên 13.0, kết quả cho thấy khả năng nhận ra sự thay đổi file nhạc rất thấp, ngoài ra với việc sử dụng một mixer tốt làm cho đoạn thông điệp gần như không thể nghe được.

**Đối Sánh**

Dựa theo quyển A COMPARISON OF ECHO HIDING METHODS - Kadir Tekeli ,Rifat Asliyan - Adnan Menderes University – ISRES phát hành



****

Ta có thể thấy khi watermark chỉ một đoạn âm thanh, cộng với tỉ lệ nhúng thấp do thông tin cần đánh dấu nhỏ, tỉ lệ SNR rất cao việc phát hiện ra dữ liệu bị ẩn thấp. Tỉ lệ BER cũng thấp trong ngưỡng cho phép có thể tự sửa lỗi bằng mã sửa lỗi Hamming

1. **TÍCH HỢP HỆ THỐNG**

4.1) Nghiệp vụ chính của hệ thống  
+ User upload bài hát  
+ Server kiểm tra bài hát đã watermark hay chưa  
+ Nếu bài hát chưa watermark thì server tiến hành watermark bài hát và lưu vào database  
+ Nếu bài hát đã watermark rồi thì server sẽ update số lần Reup của User, Reup 3 lần hệ thống sẽ tự động khóa tài khoản của User

4.2) Hệ thống có 2 phân quyền  
 - User  
 + Chưa đăng nhập  
 + Đã đăng nhập  
 - Admin

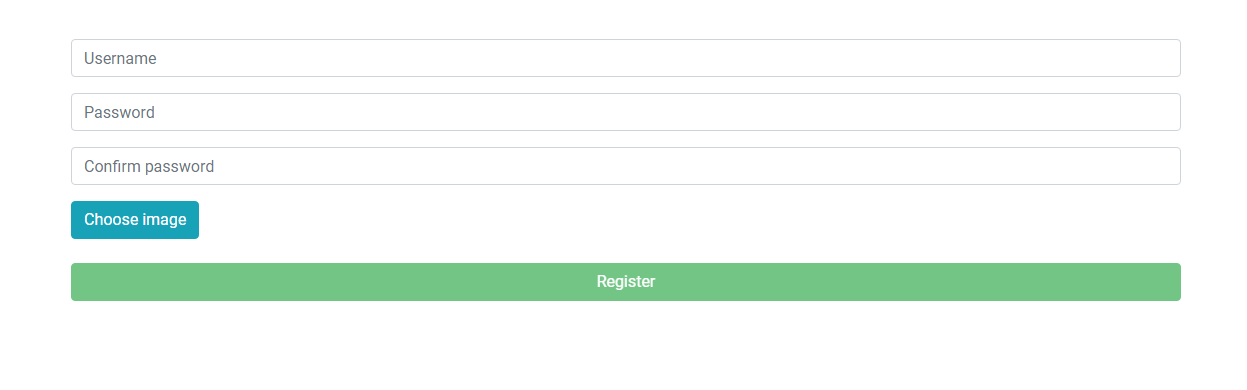
4.3) Quyền User khi chưa đăng nhập  
 + Nghe nhạc  
 + Download nhạc  
 + Xem thông tin bài hát ( chưa làm )

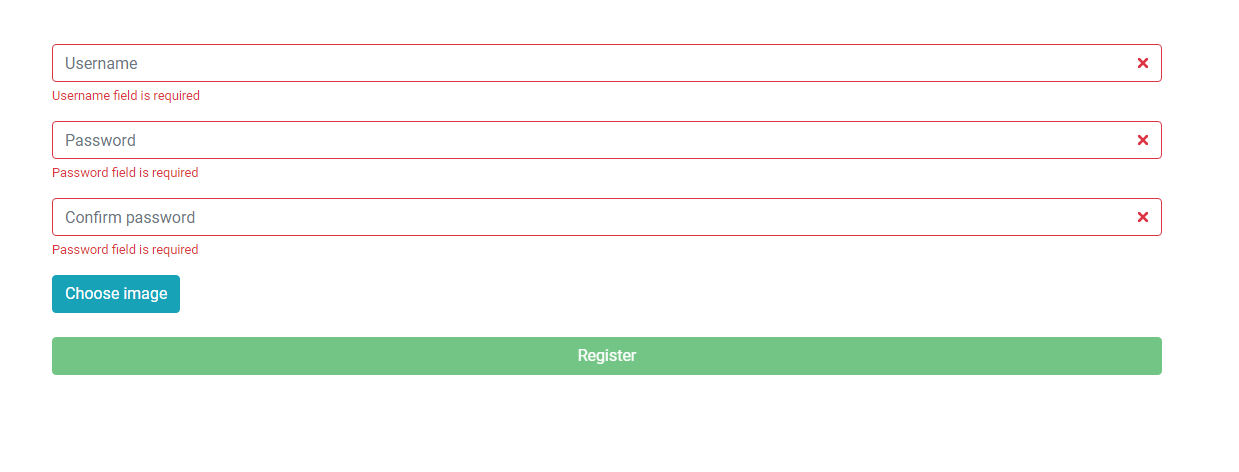
4.4) Quyền User khi đã đăng nhập  
 + Nghe nhạc  
 + Download nhạc  
 + Upload bài hát ( bài hát được upload thành công chỉ hiển thị trong album cá nhân của User. Việc đưa bài hát vào mục thể loại, nghệ sĩ sẽ do Admin quyết định )  
 + Quản lý bài hát cá nhân  
 + Edit thông tin tài khoản  
 + Nghe nhạc theo album cá nhân  
 + Nghe nhạc theo danh sách bài hát yêu thích  
 + Quản lý bài hát đã chặn  
 + Quản lý playlist cá nhân: thêm, sửa, xóa playlist  
 + Nghe nhạc theo playlist cá nhân ( đang làm )  
 + Like bài hát, Unlike bài hát, Block bài hát, Unblock bài hát  
 + Thêm, Xóa bài hát ra khỏi playlist cá nhân ( đang làm )  
 + Xem thông tin bài hát ( chưa làm )  
 + Bình luận bài hát ( chưa làm )

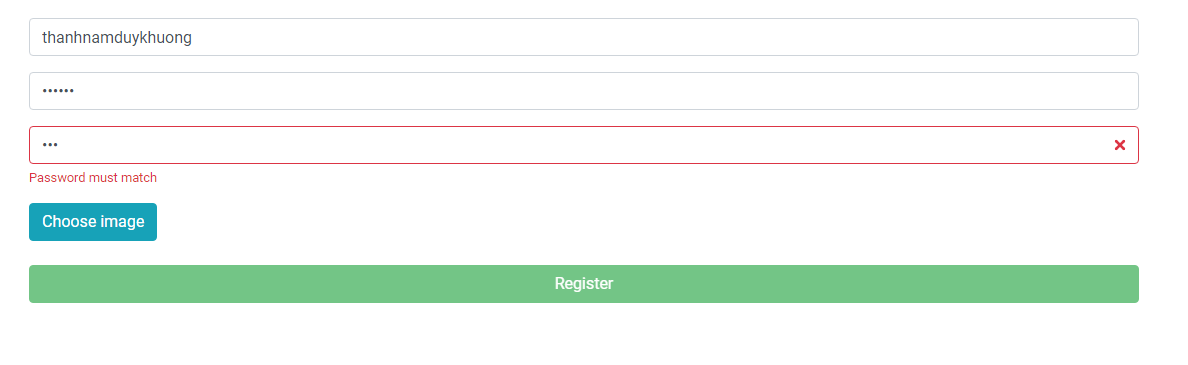
4.5) Quyền Admin  
 + Quản lý thể loại bài hát : thêm, sửa, xóa thể loại bài hát  
 + Quản lý danh sách nghệ sĩ : thêm, sửa, xóa nghệ sĩ  
 + Quản lý bài hát do User upload: Thêm, sửa, xóa bài hát. Update bài hát vào các mục thể loại, nghệ sĩ phù hợp. Nghe thử bài hát, download bài hát  
 + Quản lý User: Block, Unblock tài khoản User

4.6) Màn hình của hệ thống

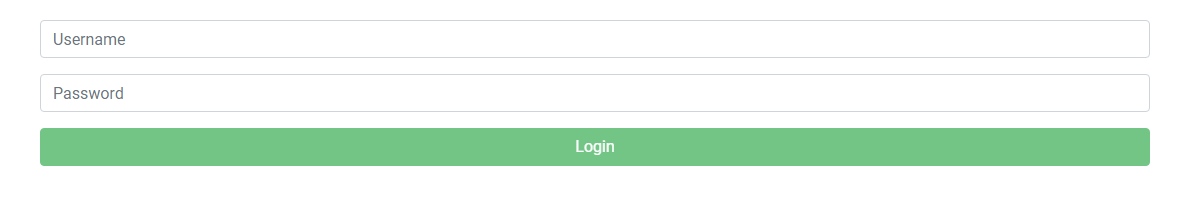
1. Màn hình đăng ký tài khoản





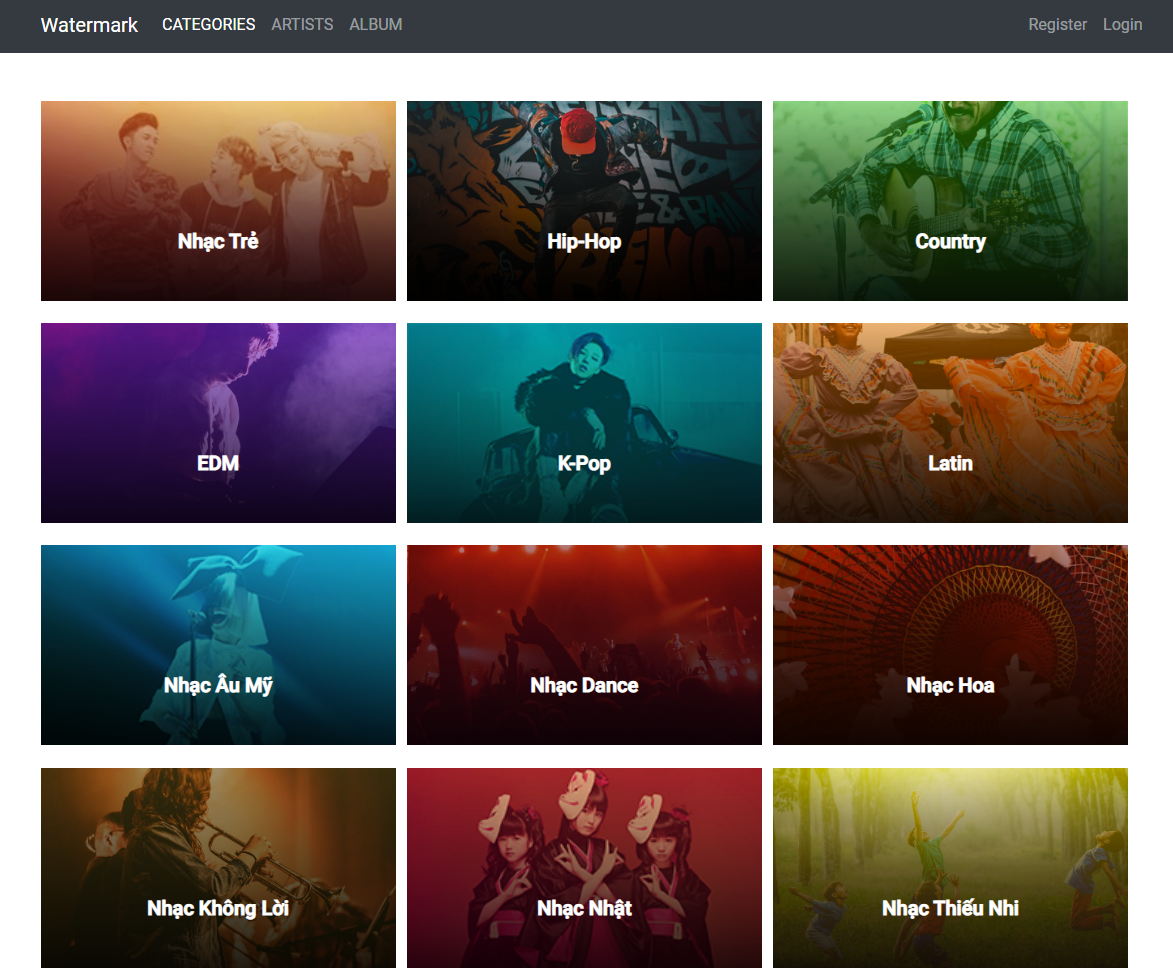


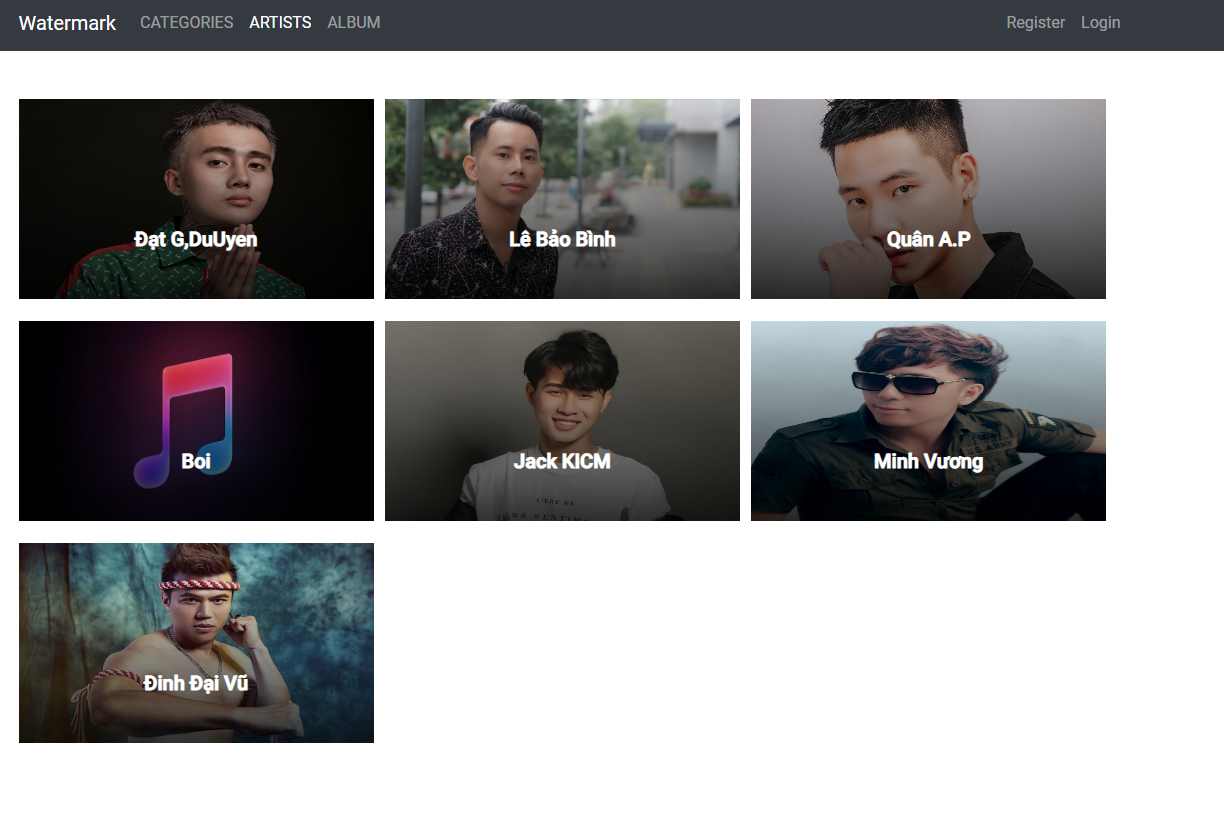
1. Màn hình đăng nhập

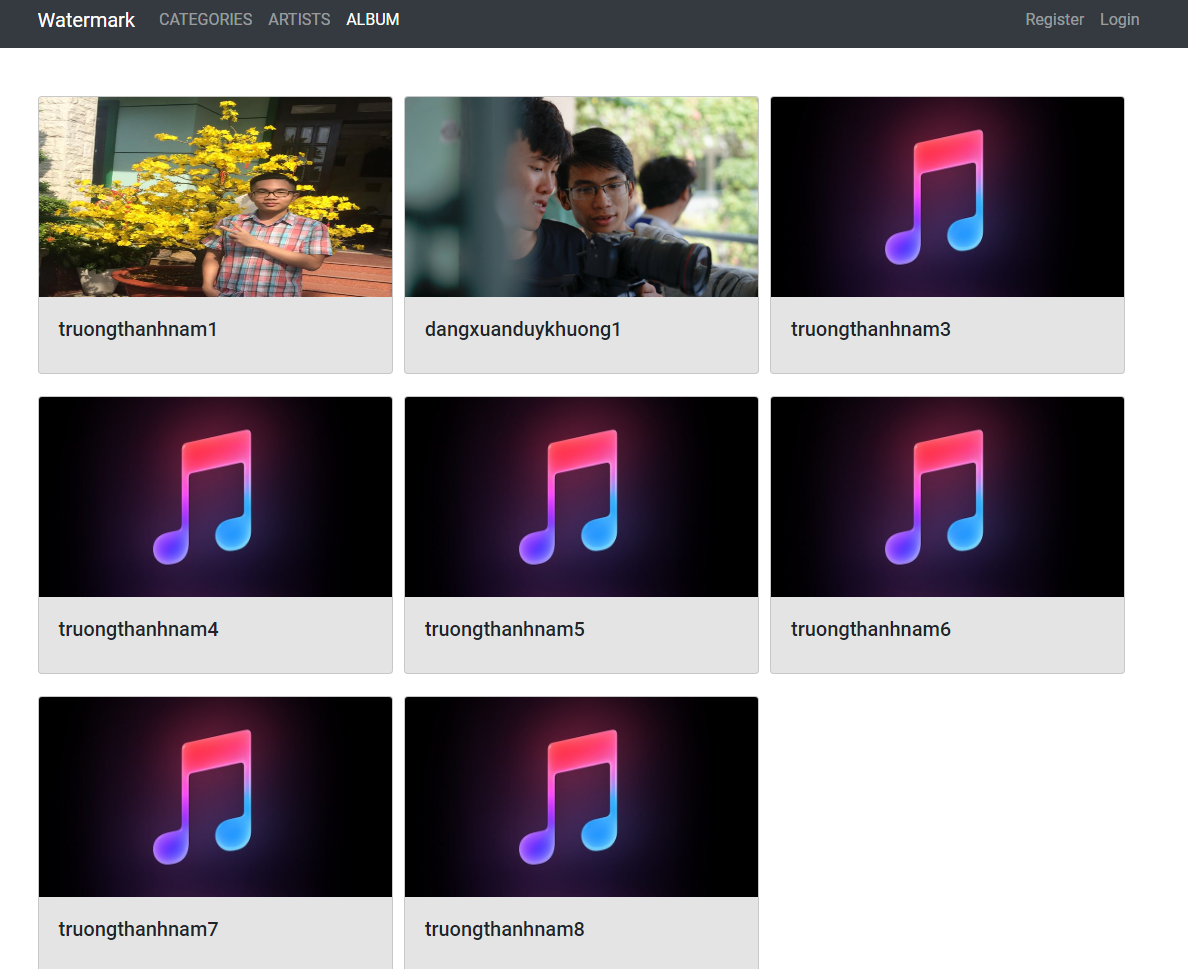




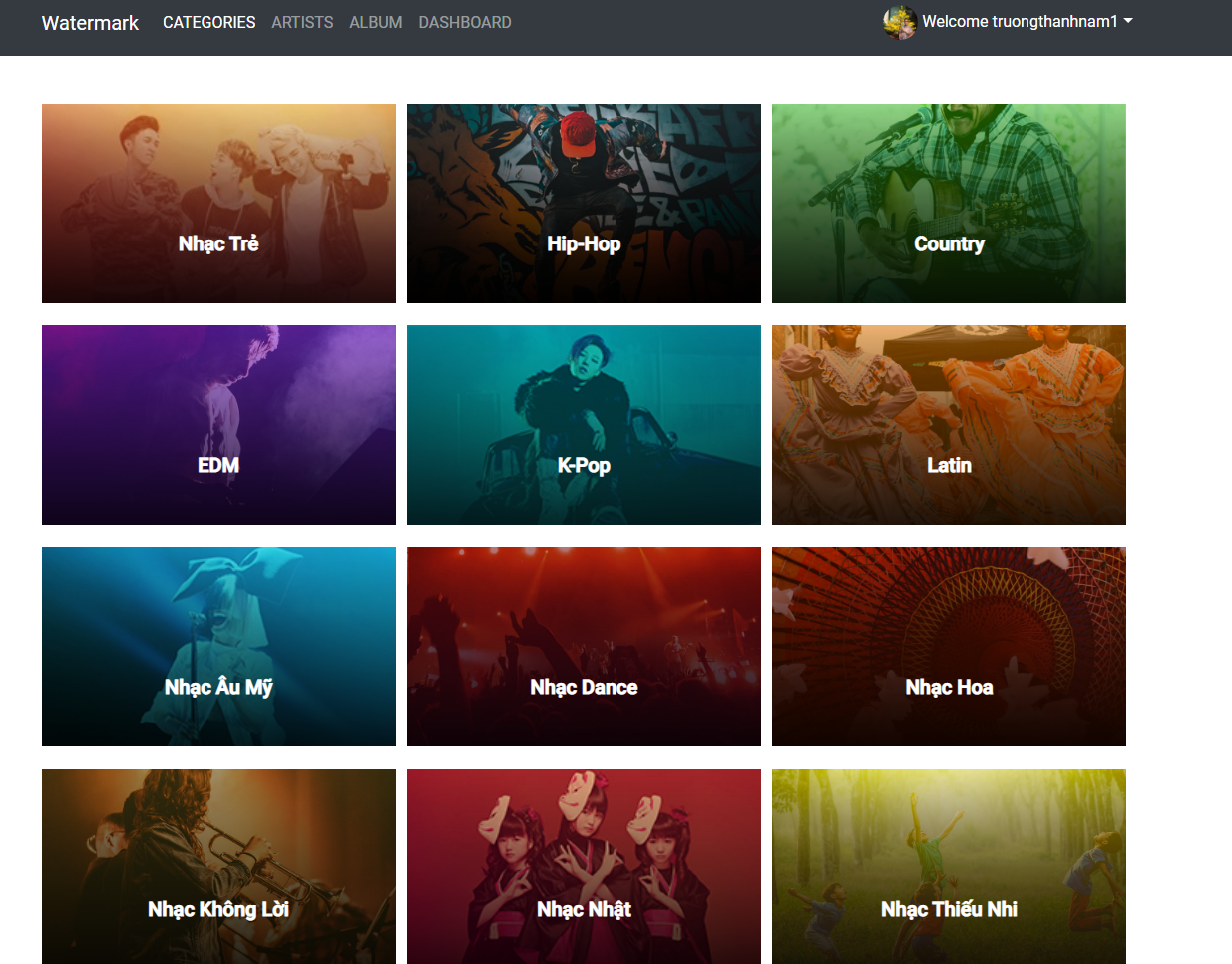
1. Màn hình thể loại nhạc



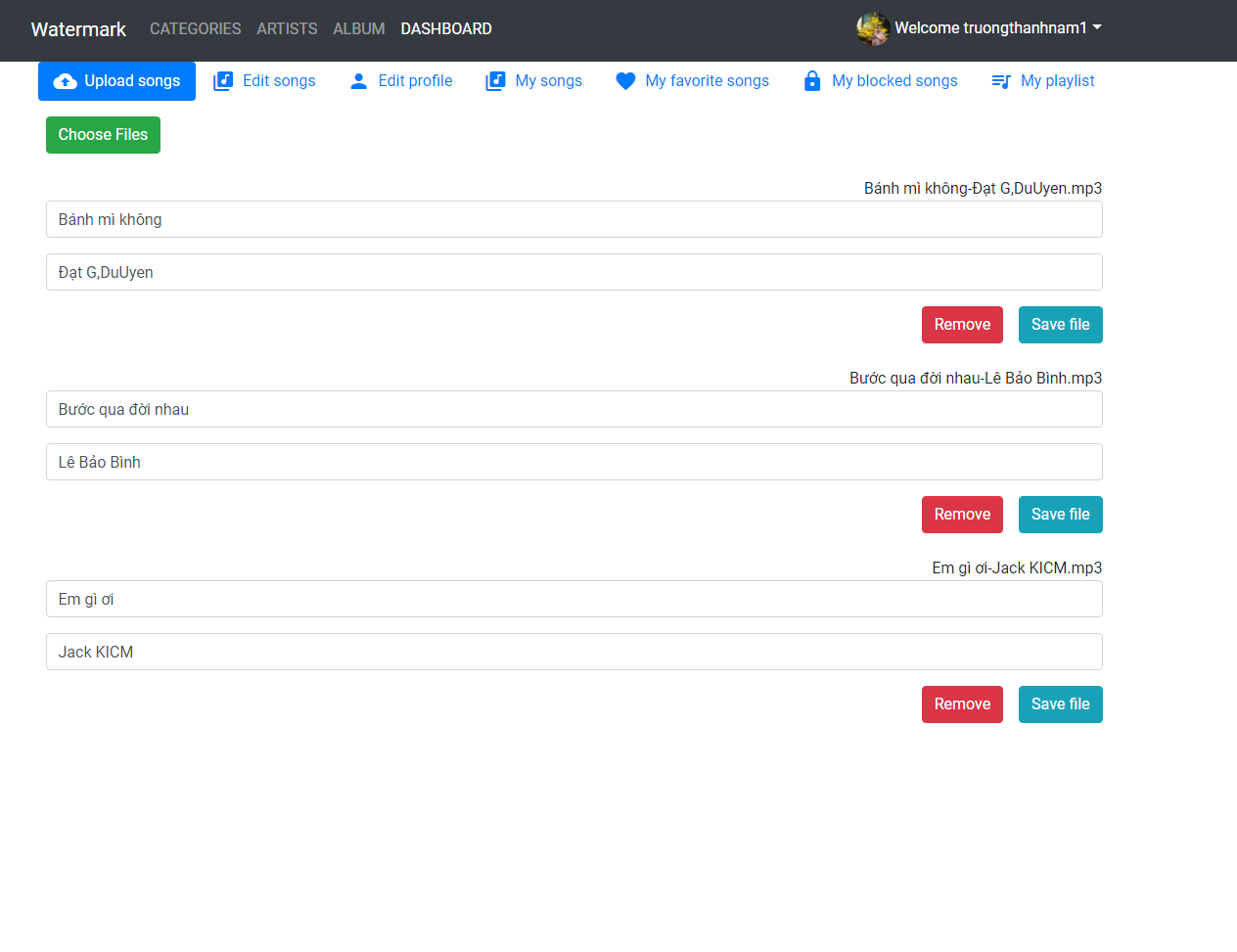
1. Màn hình nghệ sĩ
2. Màn hình Album



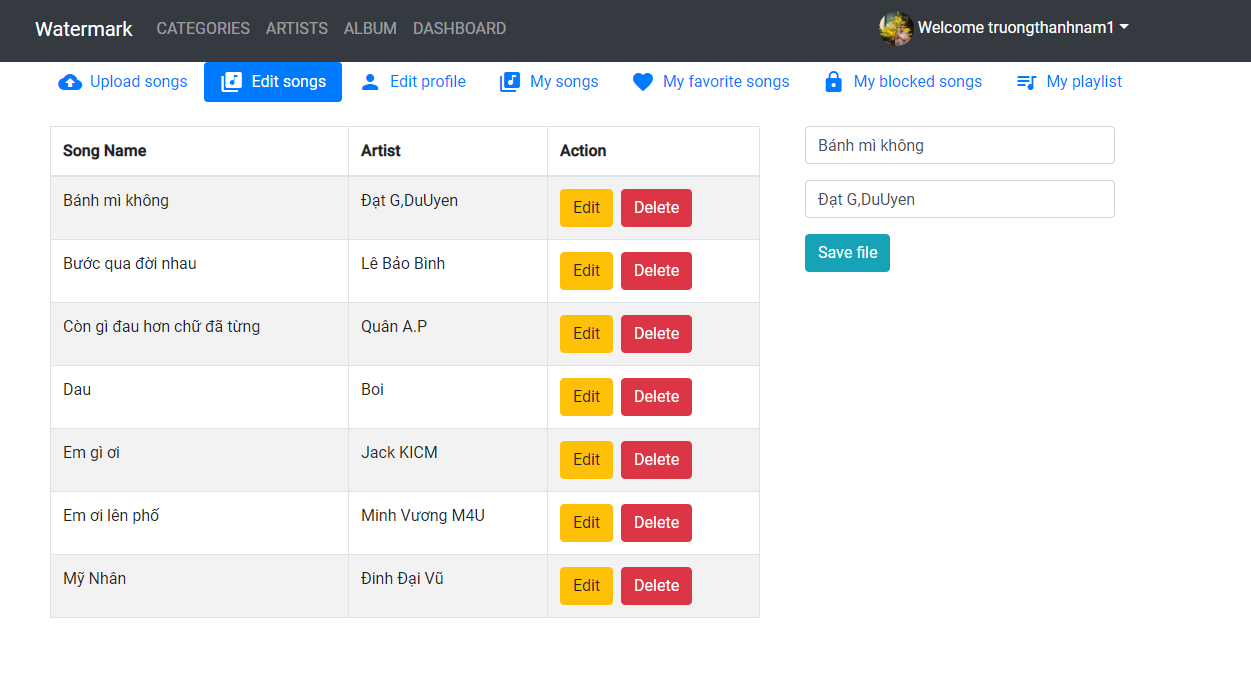
1. Màn hình phân quyền User sau khi đăng nhập



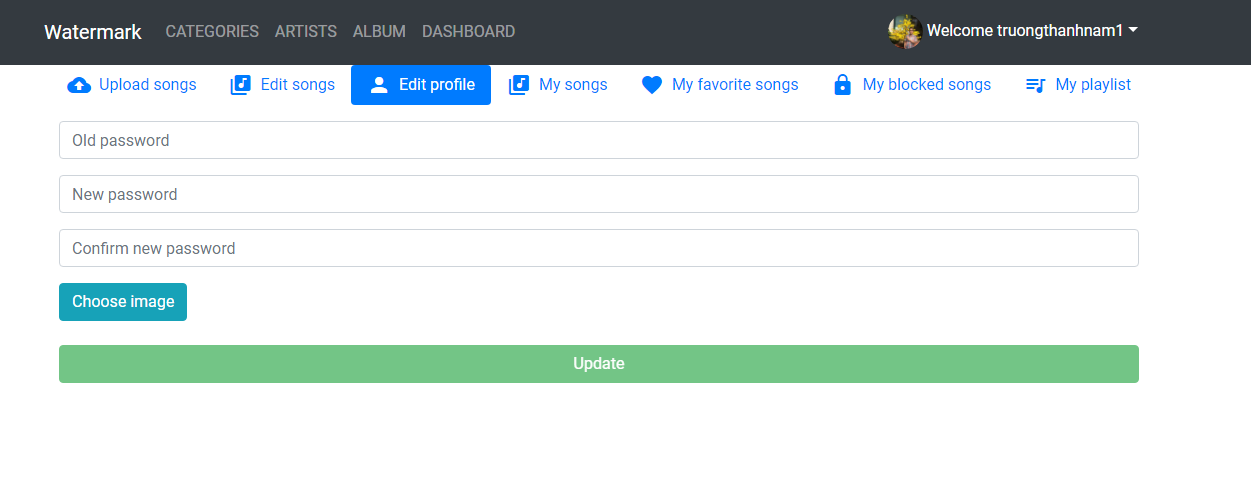
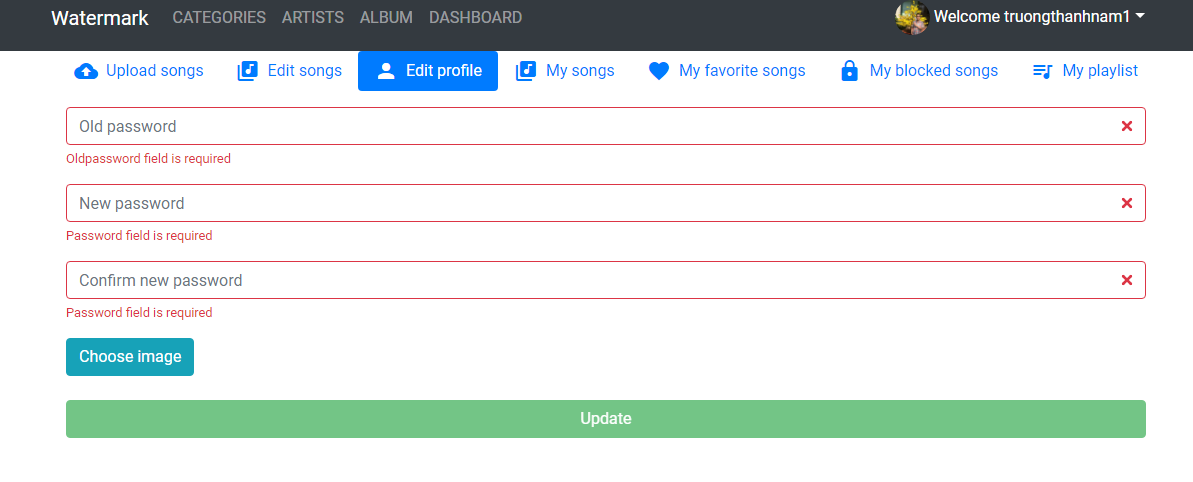
1. Màn hình Upload bài hát của User

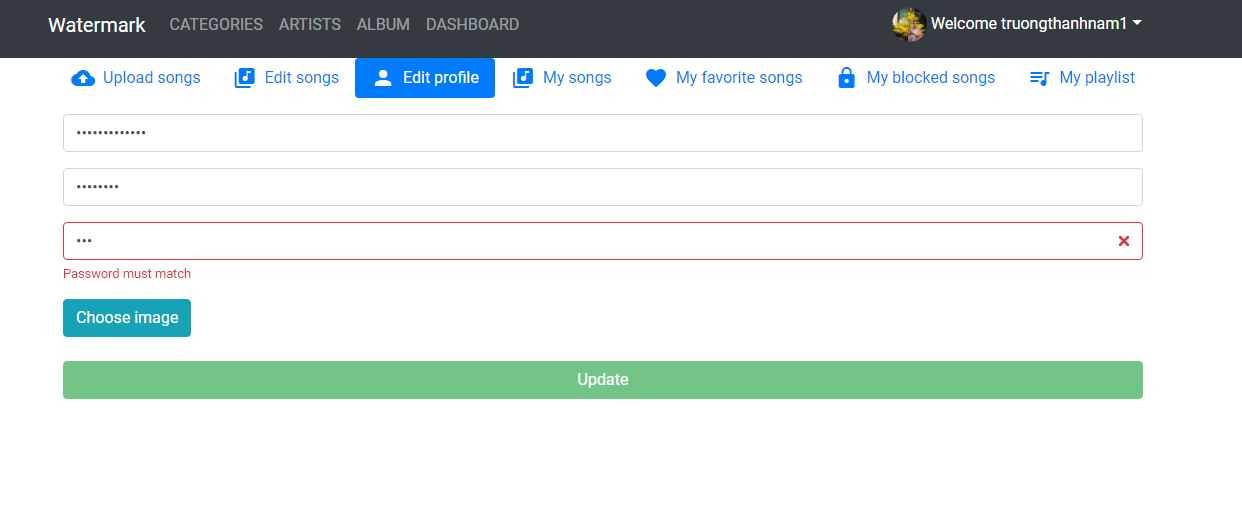


1. Màn hình Edit bài hát của User

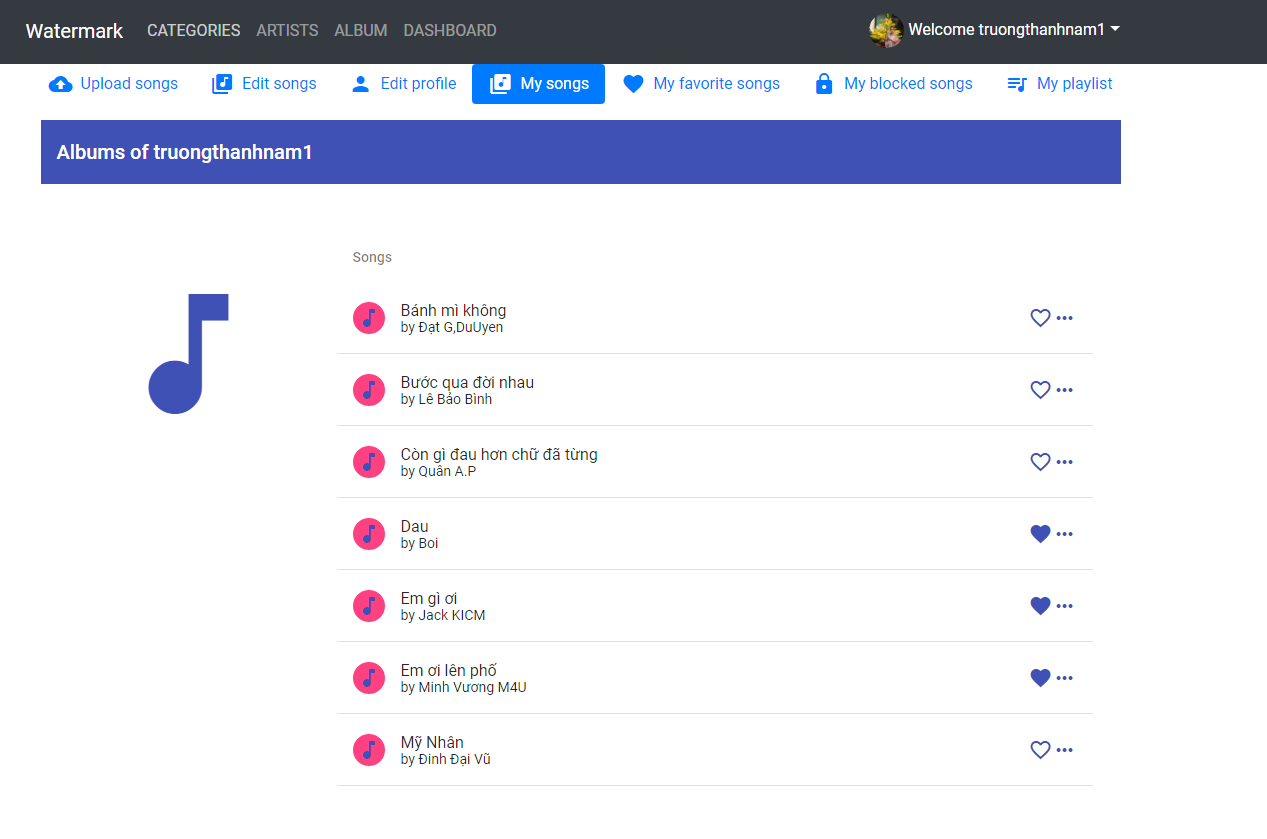


1. Màn hình Edit thông tin tài khoản

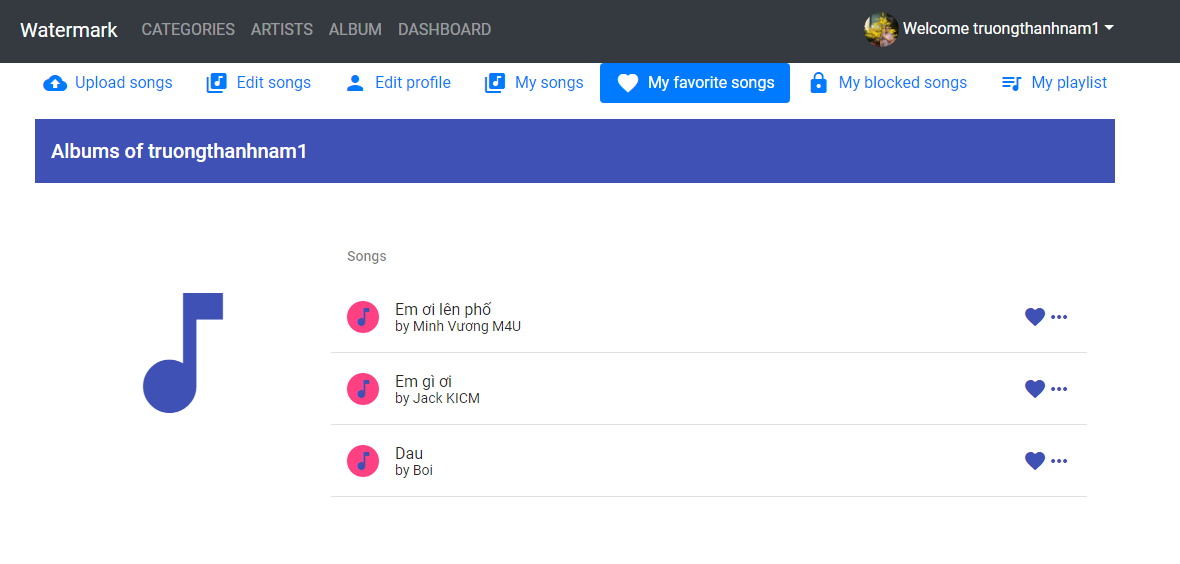
 



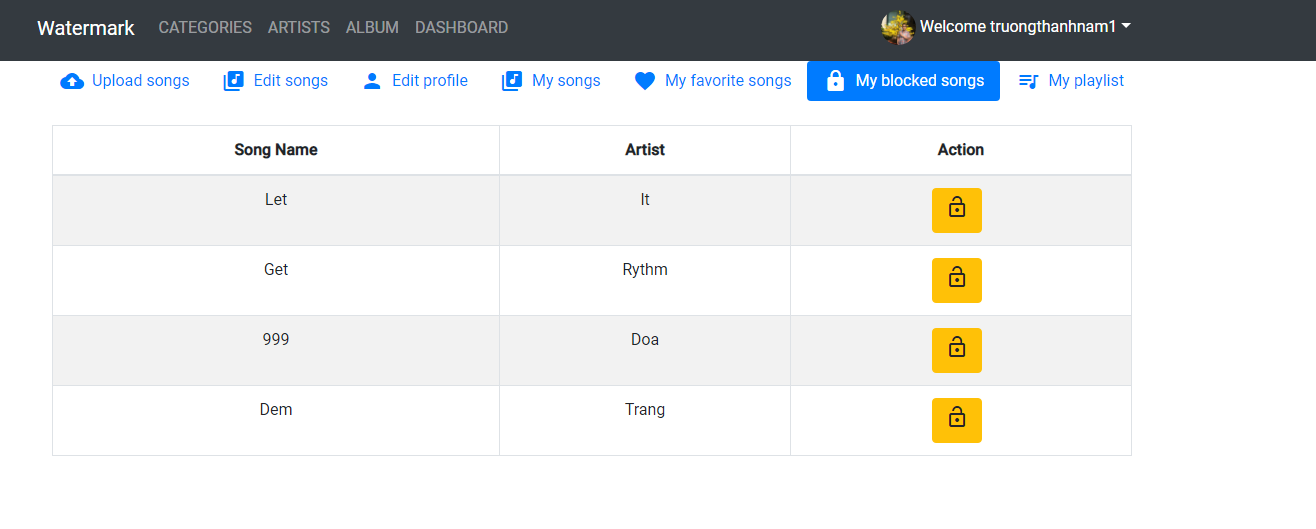
1. Màn hình hiển thị bài hát đã upload của User



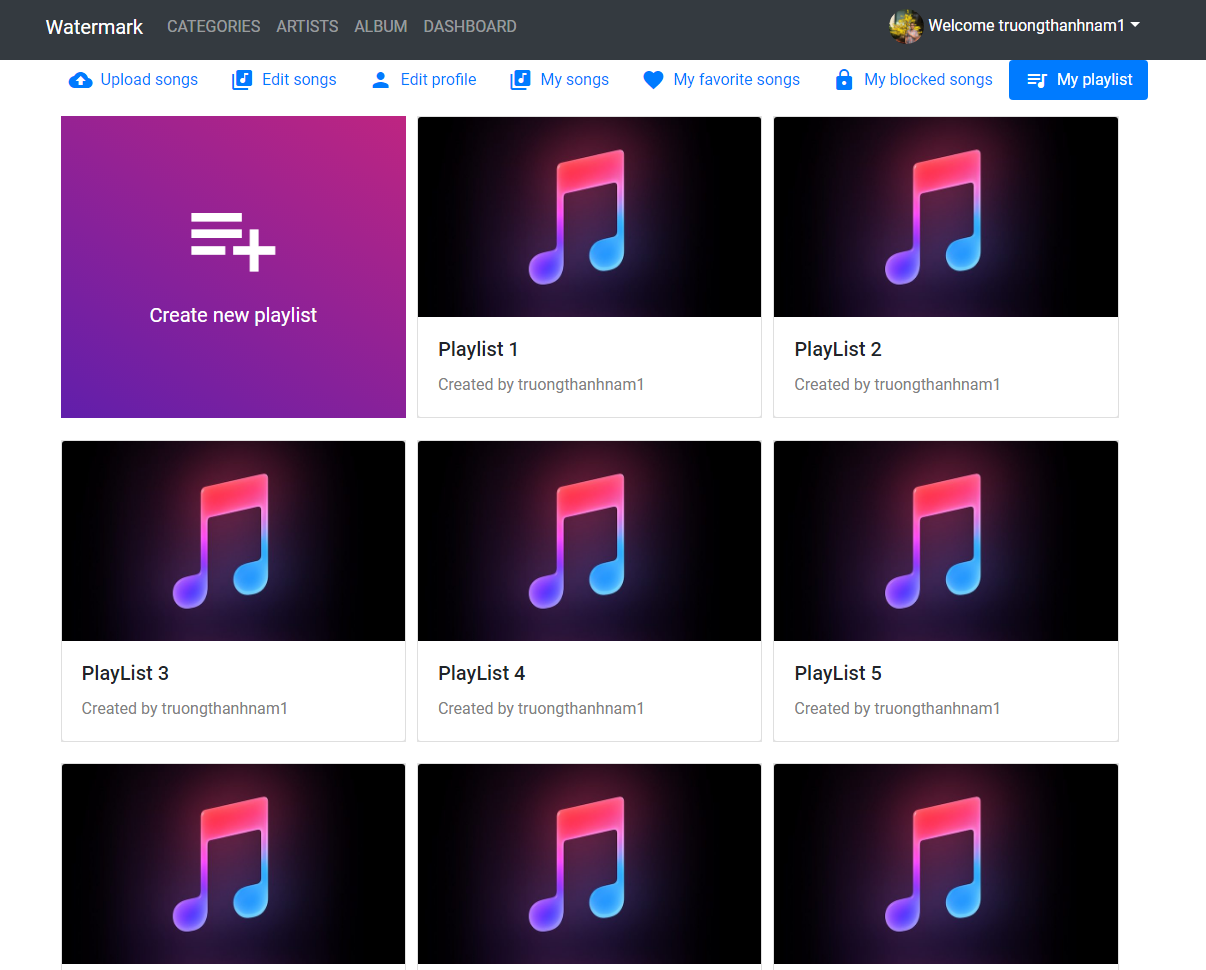
1. Màn hình hiển thị bài hát yêu thích của User

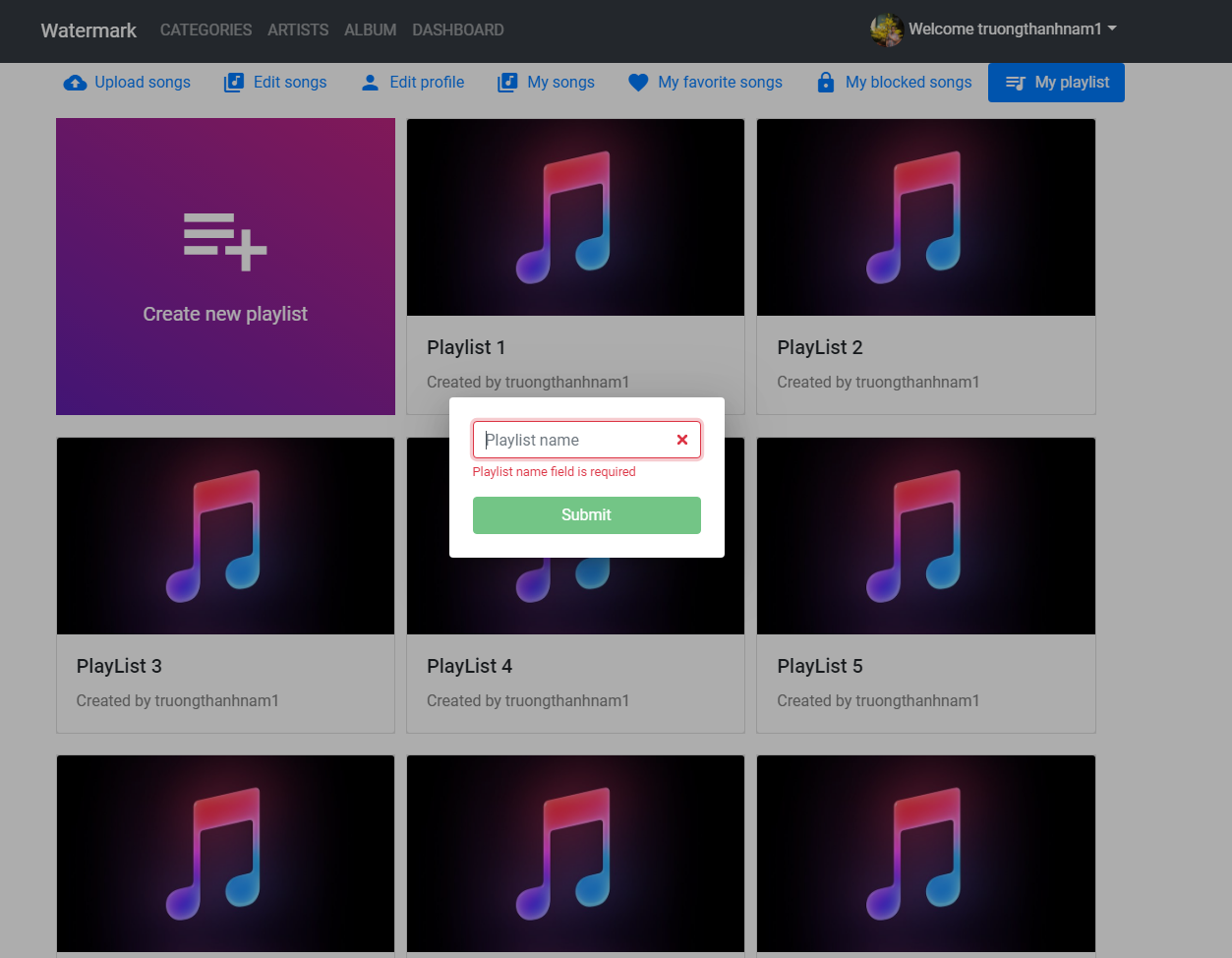


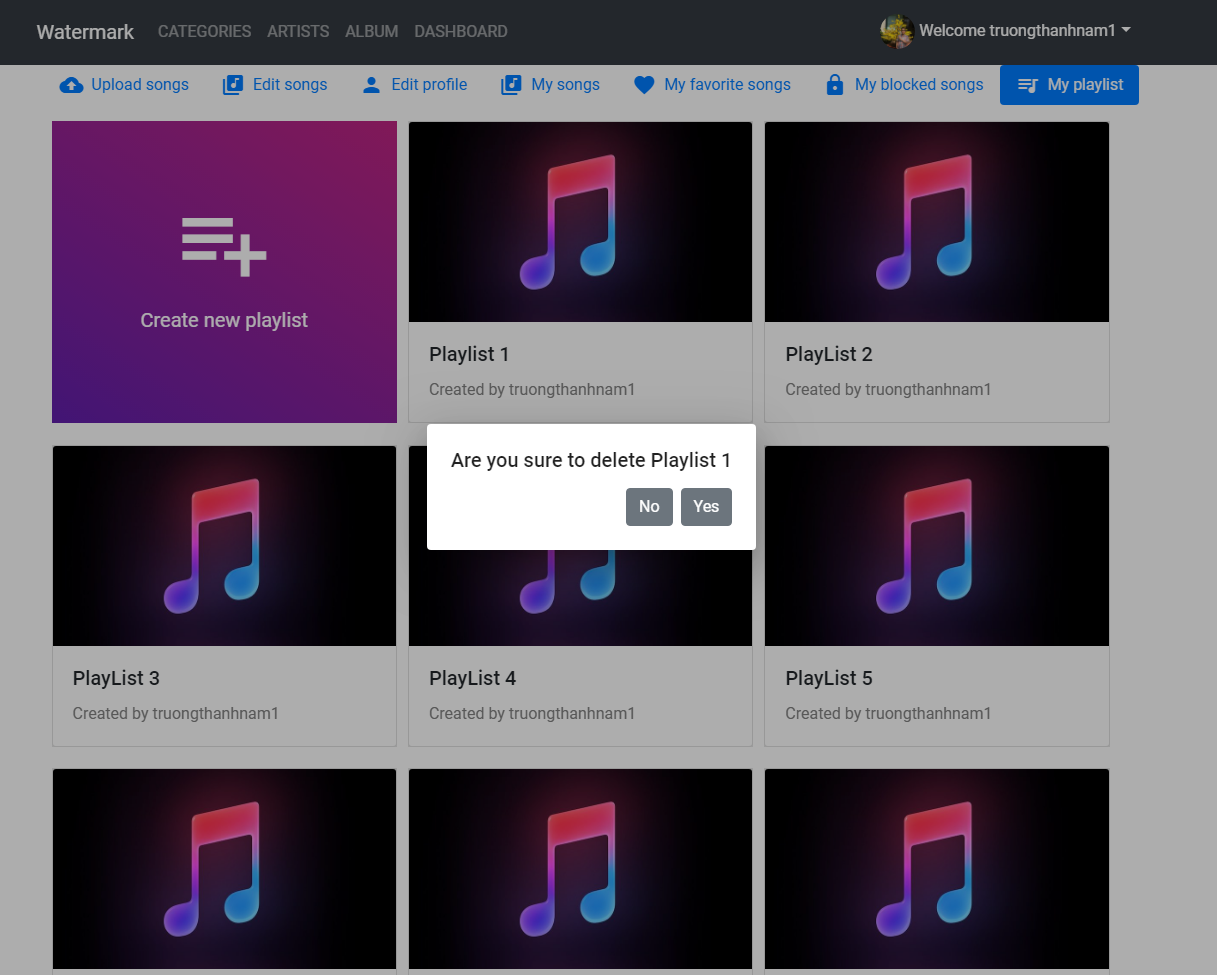
1. Màn hình hiển thị danh sách các bài hát đã block



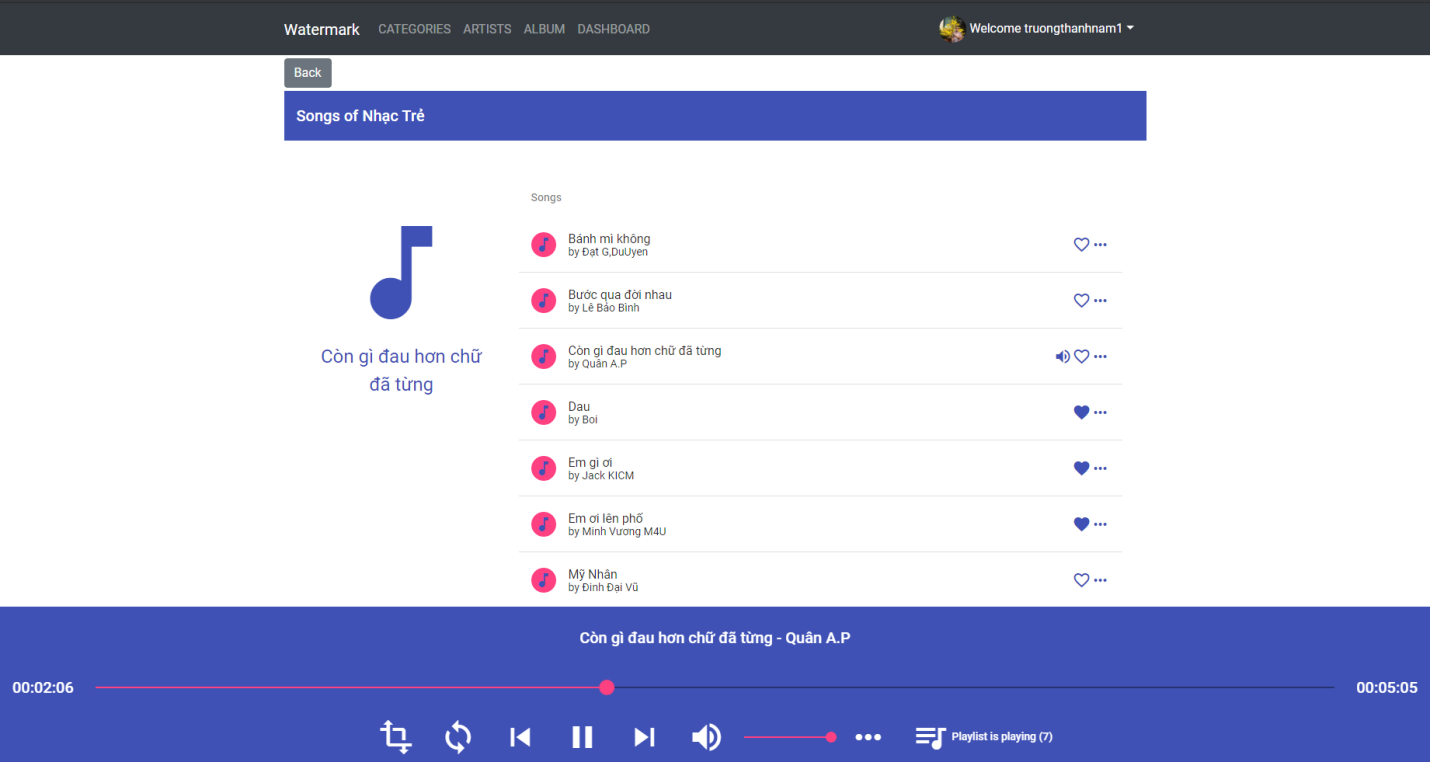
1. Màn hình hiển thị playlist cá nhân của User

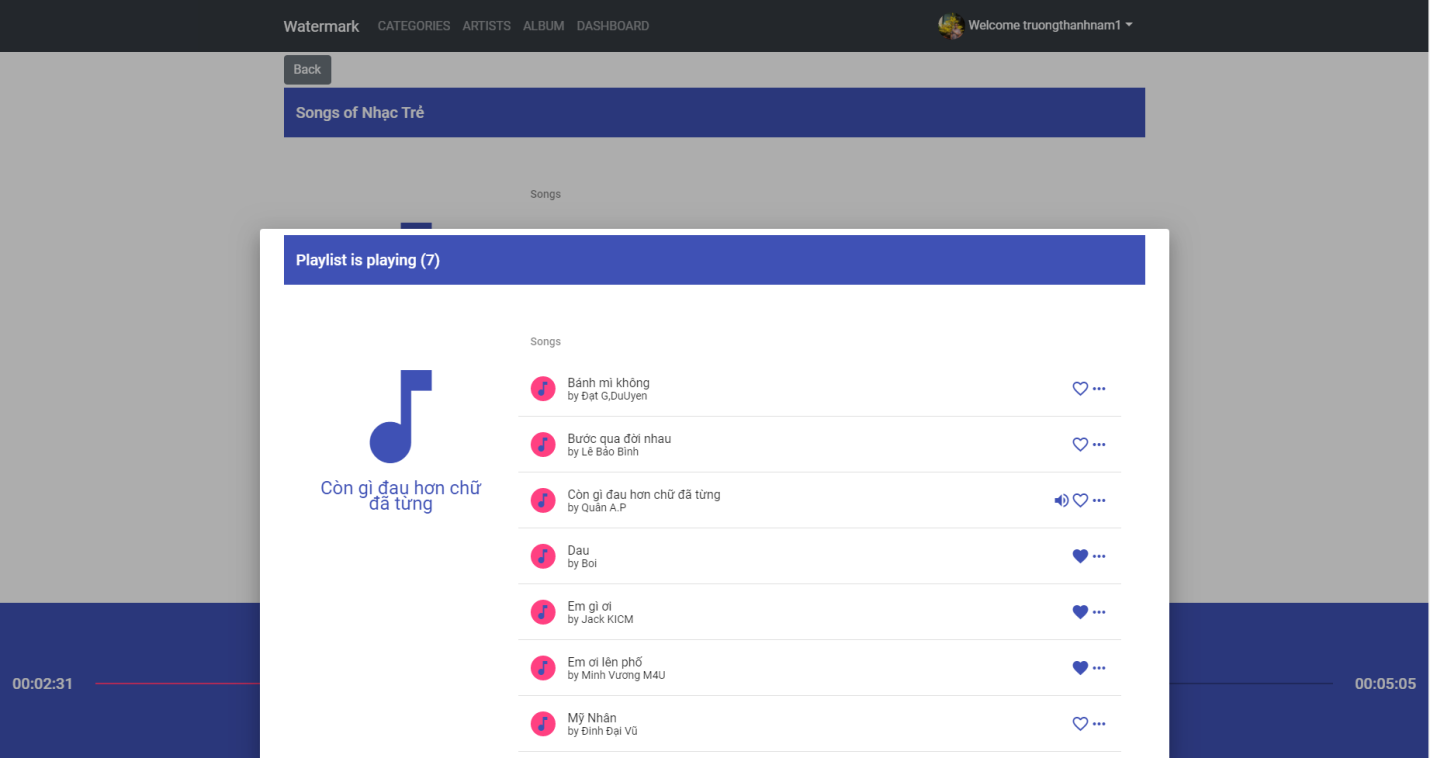






1. Màn hình khi User nghe nhạc

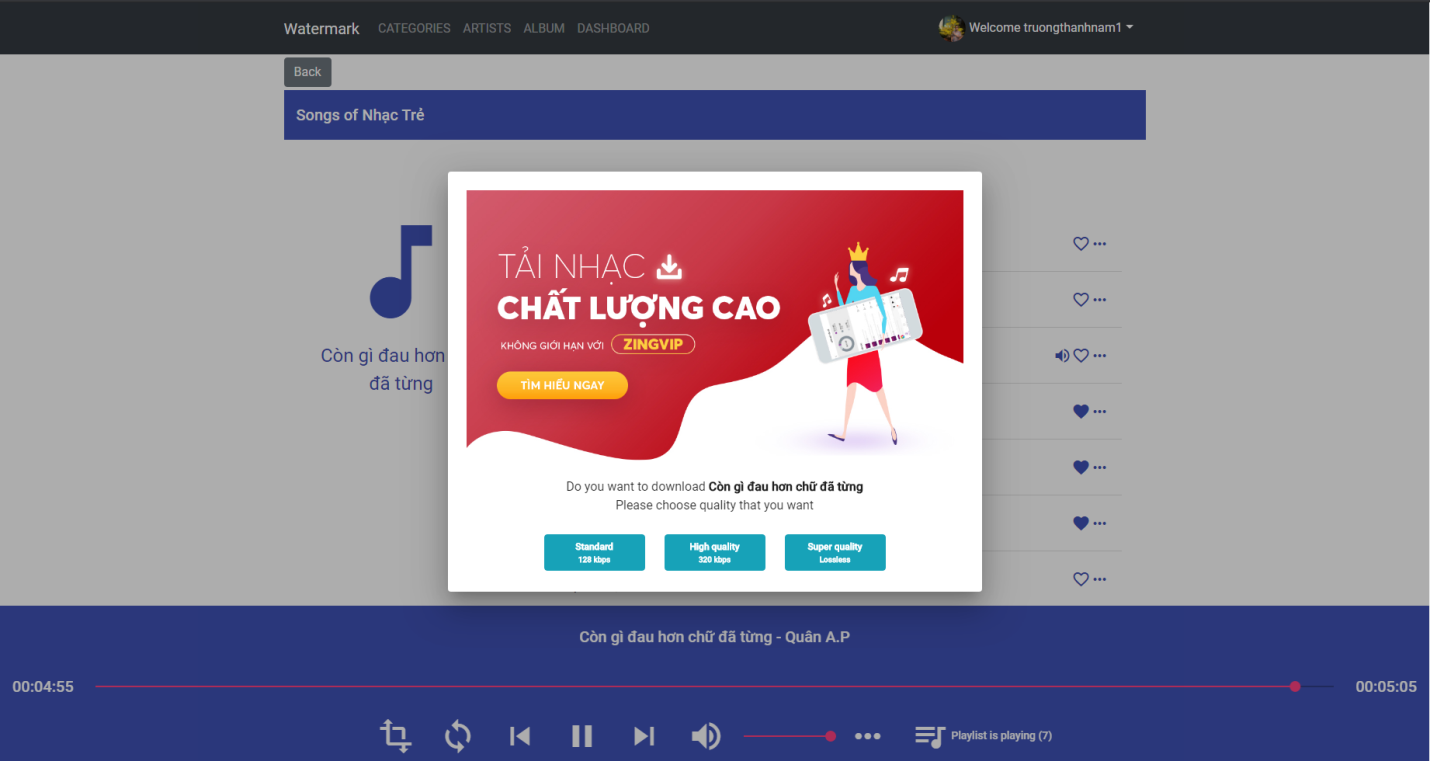




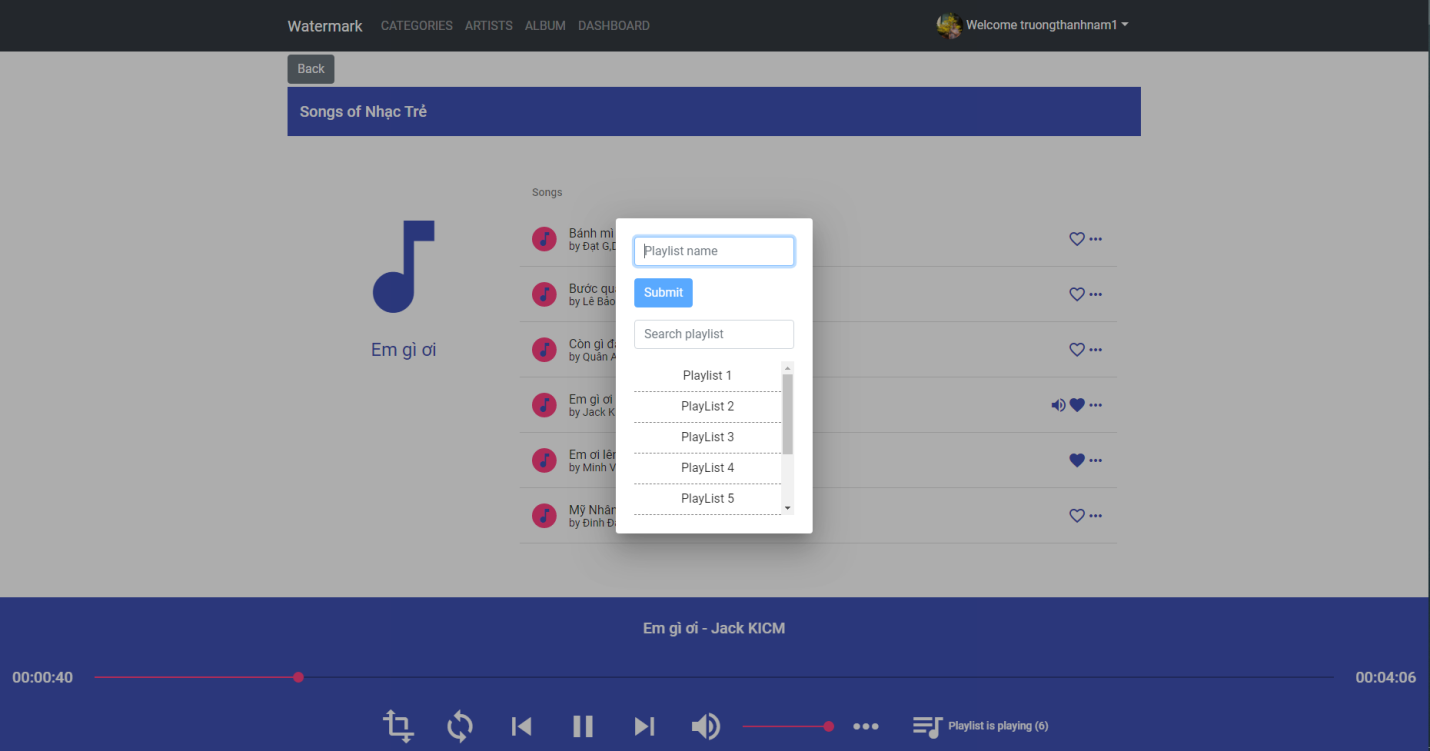
1. Màn hình khi User click vào nút 3 chấm ứng với mỗi bài hát



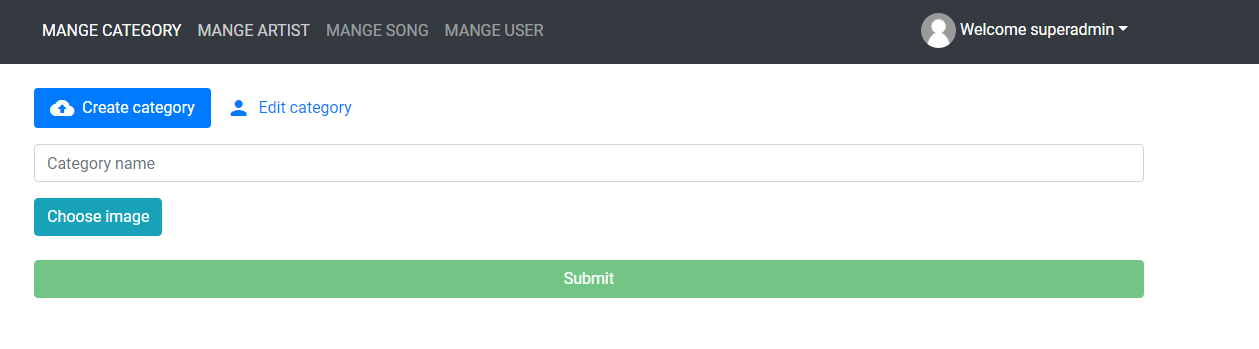
1. Màn hình khi download bài hát

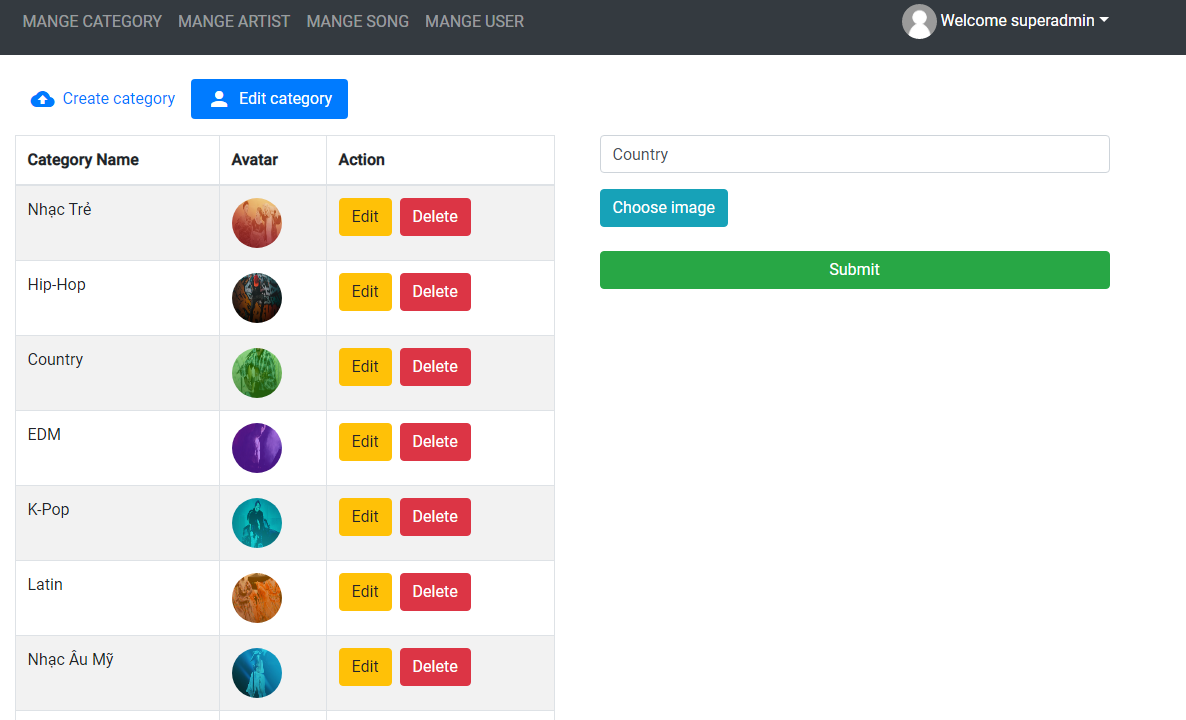


1. Màn hình khi muốn thêm bài hát vào PlayList

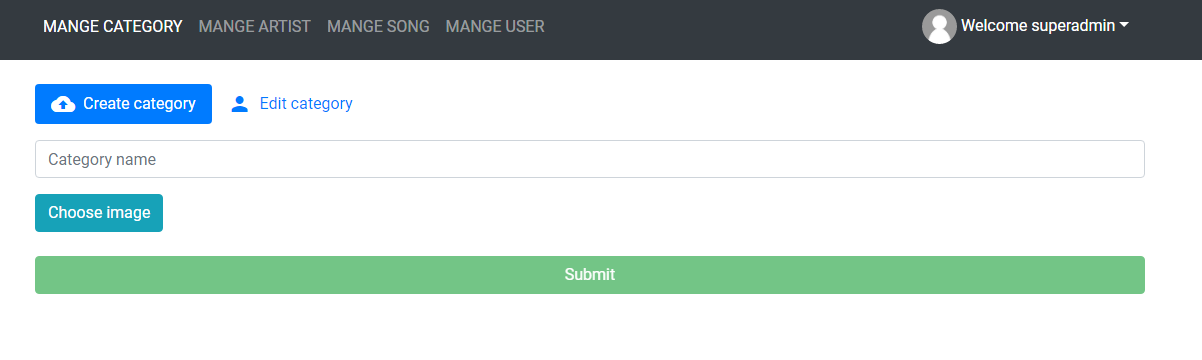


1. Màn hình quản lý thể loại bài hát

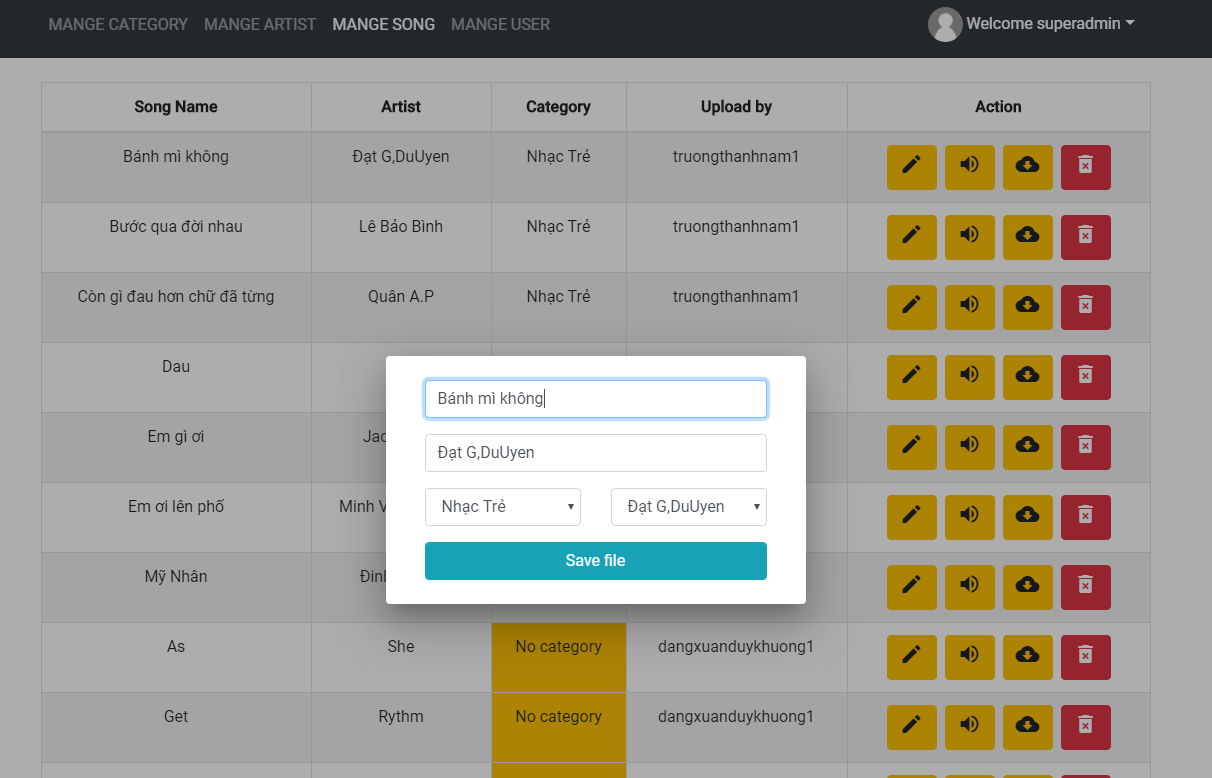
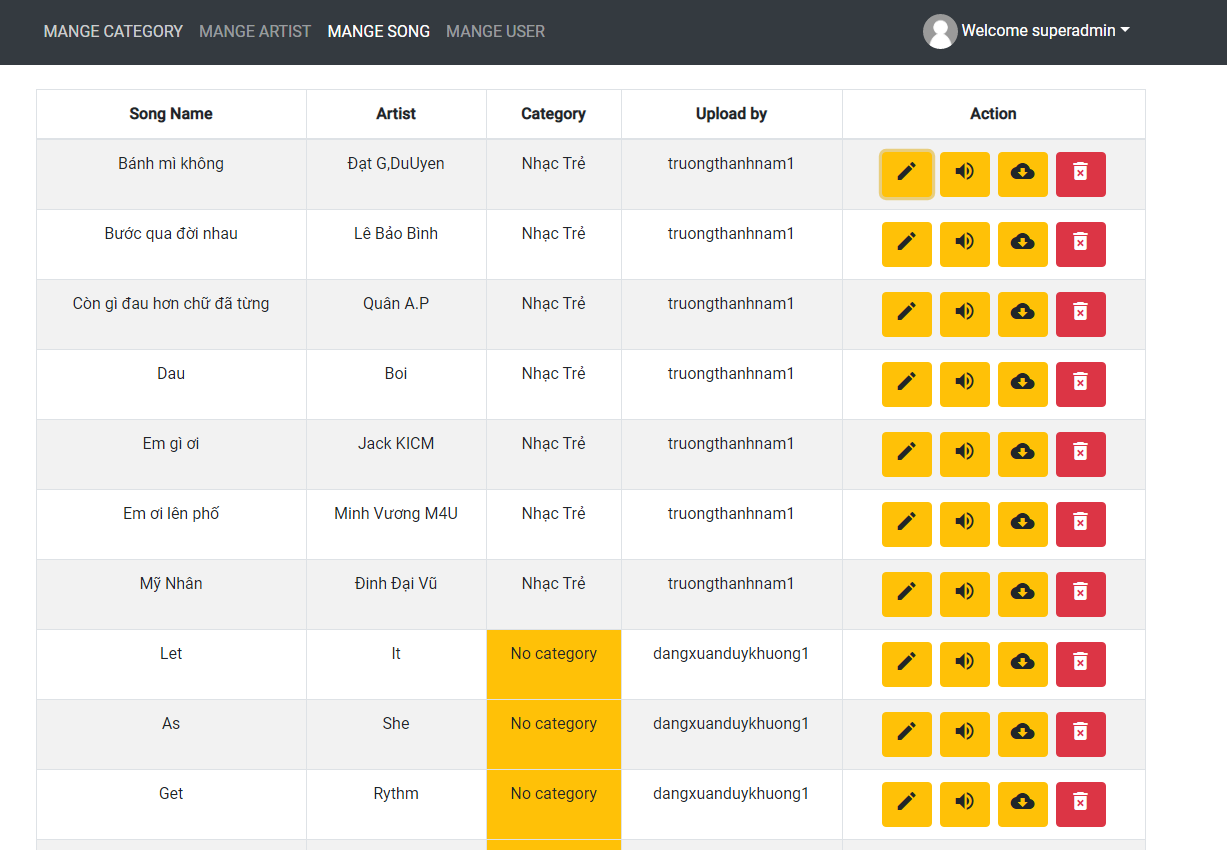


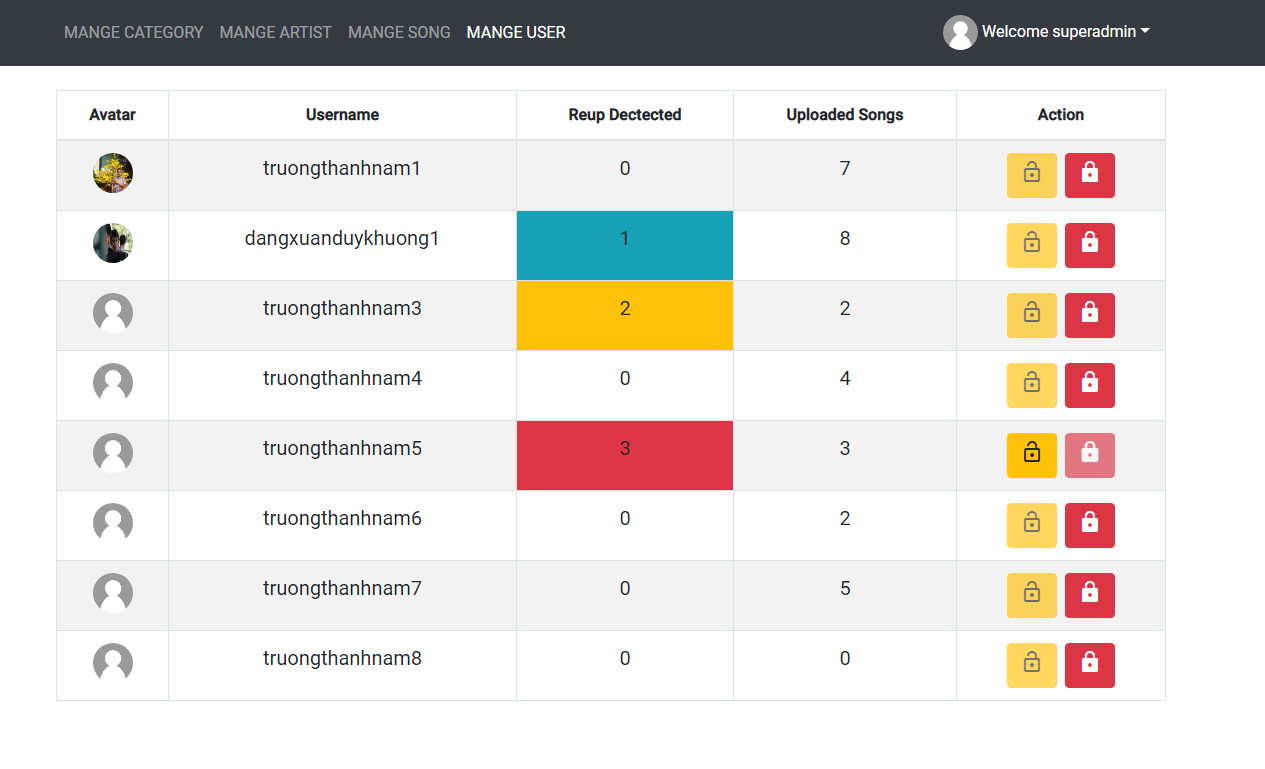


1. Màn hình quản lý nghệ sĩ





1. Màn hình quản lý bài hát
2. Màn hình quản lý tài khoản của hệ thống



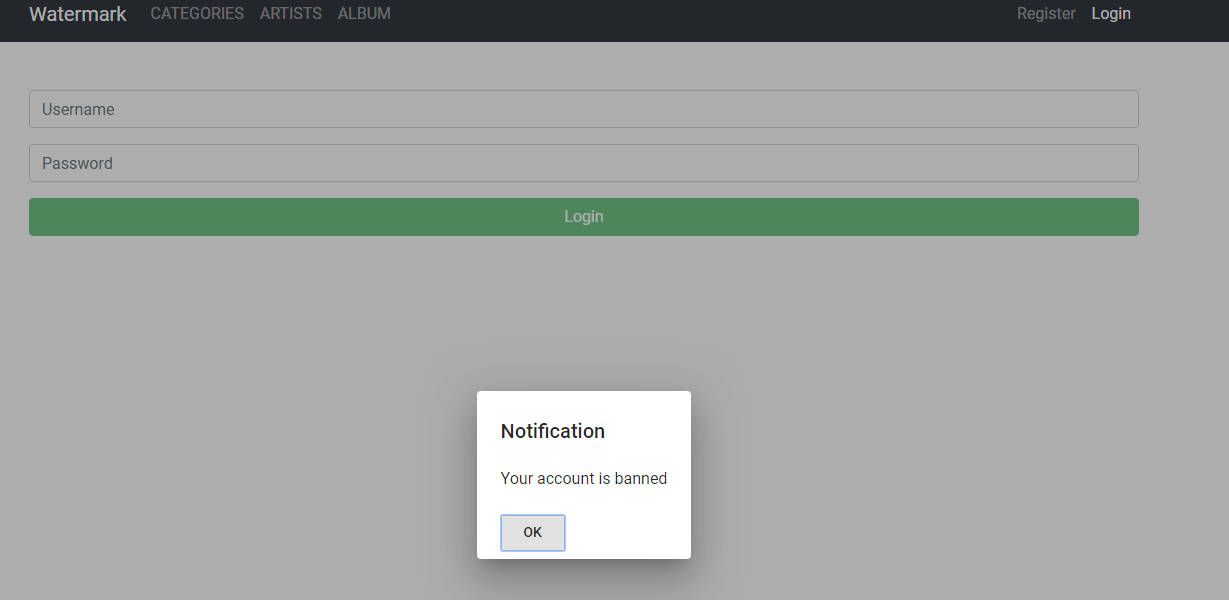
1. Màn hình hiển thị số lần Reup của User



1. Màn hình hiển thị số bài hát upload đang được server xử lý



1. Màn hình thông báo tài khoản đã bị khóa sau khi Reup 3 lần



TÀI LIỆU THAM KHẢO

1. A COMPARISON OF ECHO HIDING METHODS - Kadir Tekeli & Rifat Asliyan - Adnan Menderes University