Music Review

Develop a music review site where users can submit reviews of musical performances. Functionality is similar to IMDb.

Objectives:

A. Apply knowledge of server-side and client-side scripting and modern web application frameworks to create a complex web application.
B. Expose major functionality of the application via a ReSTful web API.
C. Develop a client application using the above API.
D. Incorporate 3rd party services to a web application.
E. Implement an authentication protocol and provide different levels of functionality to authenticated vs. unauthenticated users.
F. Implement a client application that works on both mobile as well as a variety of desktop browsers and presents a user interface that scales appropriately.
G. Develop applications that are resistant to malicious exploitation.
H. Create a security and privacy policy that is publicly accessible.
I. Create a DMCA notice & takedown policy that is publicly accessible.
J. Provide a DMCA takedown procedure and tools for the site administrator:
    a. Log requests, notices, and disputes.
    b. Send a takedown notice for any DMCA requests received and disable display of alleged copyright violations.

Description

Develop a web application for a music review site that allows browsing reviews, creating, editing and saving new reviews. Use of Node.js and Angular v8 is required. Other technologies may include Mongodb and Express or any alternatives that suit you better.

Requirements

1. Authentication method: {**total 12 points + 4 bonus points**}
    a. Local authentication mechanism which uses email as the username and require a password. {**4 points**} ❤️
    b. One external authentication mechanism (e.g. Google, Apple, Facebook etc) that authenticates via a third-party such as Google, Facebook etc. {**4 bonus points**}
    c. Input validation for email (proper format). {**2 points**} ❤️
    d. Local passwords stored using the most secure method available (e.g. Argon2). {**1 point**} ❤️
    e. Verification of of email (see References). {**3 points**}
    f. If the account is marked as deactivated, show a message asking to contact the site administrator and not allow logging in. {**2 points**}
2. Limited functionality for unauthenticated users. {**total 24 points**}
    a. Start page showing application name, a short “about” blurb that says what the site offers, and login button. {**2 points**}❤️
    b. List of songs (up to 10) ordered by popularity (e.g. number of users who reviewed that song or average rating). Any sensible criteria of popularity is acceptable. {**4 points**}❤️
    c. Ability to search songs based on keywords. {**6 points**}❤️
    d. Keywords are matched with all attributes of the item (see ID3v1 below). {**2 points**} ❤️
    e. Keywords are soft-matched (e.g ignore differences in case, white-space, minor spelling variations). {**2 points**}

    f. Ability to view all information on a song by clicking or expanding (shows all attributes, most recent review, the number of reviews and the average rating. {**4 points**}
    g. Ability to view all reviews for a song: {**2 points**}
    h. Each review shows the rating, the review and reviewer's username: {**2 points**}
3. Additional functionality for authenticated users: {**total 18 points**}
    a. Add a review to a song. {**4 points**}
    b. Add a rating (1-5, star etc) to reviews created by the user. {**2 points**}
    c. Add a new song to the site. {**4 points**}
    d. Support storing all ID3v1[1] attributes for each new song. {**4 points**}
    e. Enforce required attributes "title" and "artist" when adding a new song. {**2 points**}
    f. Add a review while adding a new song if necessary. {**2 points**}
4. Site manager functionality related to site maintenance: {**total 10 points**}
    a. Special user with site manager access. {**4 points**}
    b. Ability to grant site manager privilege to one or more existing users: {**2 points**}
    c. Ability to mark a song as hidden and clear the "hidden" flag if set: {**2 points**}
    d. Ability to mark a user as "deactivated" and mark as "active" if deactivated: {**2 points**}
5. Web service API: {**total 14 points**}
    a. Provide an API that has at least four nouns (URLs) and at least 8 separate noun+ HTTP verb combinations. Must have at least two nouns that support two or more verbs. {**8 points**}
    b. Build your application using this API. {**6 points**}
6. Site manager functionality related to copyright enforcement: {**total 12 points**}
    a. Create a security and privacy policy that is publicly accessible. {**2 points**}
    b. Create a DMCA notice & takedown policy that is publicly accessible. {**2 points**}
    c. Provide a DMCA takedown procedure and tools for the site administrator: {**total 8 points**}
        i. Document to describe the workflow and usage of tools. {**2 points**}
        ii. Tools to log requests, notices, and disputes. E.g. Set-up properties for storing "date request received", "date notice sent", "date dispute received" with each song and provide an interface to set these properties. {**2 points**}
        iii. Tools to disable display of songs with alleged copyright violations. {**2 points**}
        iv. Tools to restore any contested songs. {**2 points**}

References

1. Authentication library: http://www.passportjs.org/ or https://auth0.com/
2. Email verification: https://www.npmjs.com/package/email-verification
3. Salted Password Hashing - Doing it Right: https://crackstation.net/hashing-security.htm
4. Responsive design using Angular - https://material.angular.io/
5. DMCA Demystified: http://www.sfwa.org/2013/03/the-dmca-takedown-notice-demystified/
6. GitHub DMCA policy: https://help.github.com/articles/dmca-takedown-policy/
7. Angular Security; https://angular.io/guide/security

Resources

1. Firebase is a good option which provides authentication and database: https://firebase.google.com
2. JWT is the recommended method for implementing authentication and protected routes:
    a. Main site: https://jwt.io
    b. Good tutorial: https://github.com/dwyl/learn-json-web-tokens
3. Copyright enforcement functionality: See slides 16-19 of "social issues" unit

[1] See https://en.wikipedia.org/wiki/ID3