

Media Streaming Website with IBM Cloud

Video streaming

Project Overview

This project aims to create a media streaming website that allows users to upload, view, and share video content. The website integrates with IBM Cloud Video Streaming to provide a high-quality video playback experience. The central feature of the project is video uploading, enabling users to contribute their own content to the platform.

Table of Contents

1. **Project Objectives**
2. **Technologies Used**
3. **Project Structure**
4. **Video Upload Feature**
5. **User Authentication**
6. **Integration with IBM Cloud Video Streaming**
7. **Database Structure**
8. **Testing and Quality Assurance**
9. **Challenges Faced**
10. **Future Improvements**
11. **Conclusion**

1. Project Objectives

- Create a user-friendly media streaming website.
- Implement a video upload feature for users to contribute their content.
- Integrate with IBM Cloud Video Streaming for video playback.
- Ensure secure user authentication and access control.
- Store video metadata and user account information in a database.
- Provide a responsive and visually appealing user interface.

2. Technologies Used

- Front-end: HTML, CSS, JavaScript
- Back-end: Node.js, Express.js
- Database: MongoDB
- Video Streaming: IBM Cloud Video Streaming

- Authentication: Passport.js
- Additional libraries: Multer, Cors

3. Project Structure

The project is divided into the following components:

- Front-end: Handles user interactions and displays video content.
- Back-end: Manages the server, database, user authentication, and video uploading.
- Database: Stores user account information and video metadata.
- IBM Cloud Video Streaming: Integrates with the platform for video playback.
- Video Upload Feature: Allows users to upload videos.
- User Authentication: Ensures secure user access and management.

4. Video Upload Feature

4.1. User Interface

- Video upload form with file input and title field.
- File upload handled by Multer library.
- Video metadata input by the user.

4.2. Back-end

- Multer configured to handle file uploads.
- Route for video upload created.
- Integration with IBM Cloud Video Streaming to upload and store videos.
- Video metadata stored in the database.

4.3. User Experience

- Users can upload videos with custom titles.
- Uploaded videos integrated into the media library.

5. User Authentication

- Passport.js library used for user authentication.
- User registration and login forms.
- Protected routes to secure video uploading and user management.

6. Integration with IBM Cloud Video Streaming

- Integration with the IBM Cloud Video Streaming API for video playback.
- Authentication and secure key management implemented.

- Video streaming requests handled seamlessly.

7. Database Structure

- MongoDB used for storing user accounts and video metadata.
- User model for account information.
- Video model for video metadata.

8. Testing and Quality Assurance

- Extensive testing for functionality, security, and user experience.
- Addressed potential vulnerabilities and issues.
- Verified video playback and upload functionality.

9. Challenges Faced

- Overcoming challenges related to video file uploads and streaming.
- Ensuring security of user accounts and video content.
- Managing complex integration with IBM Cloud Video Streaming.

10. Future Improvements

- Implement user profile management features.
- Enhance video quality and adaptive streaming.
- Implement content recommendation algorithms.
- Expand video format and codec support.
- Address performance and scalability concerns.

11. Conclusion

The media streaming website with video upload feature is a significant achievement. It provides users with a platform to upload, share, and view video content seamlessly. The integration with IBM Cloud Video Streaming ensures a high-quality video playback experience. Future improvements and enhancements will further enhance the platform's capabilities.