Media Streaming

Date	11 October 2023
Team ID	5566
Team Name	Proj_227254_Team_2
Project Name	Media streaming with IBM Cloud streaming

PHASE 2
INNOVATION

The innovation phase is where we turn our design thinking concepts into actual cloud media streaming. This paper details the detailed measures that will be done to implement the design concept.

Flask: A Flexible and Lightweight Python Web Framework :

Flask is a Python web framework that focuses on simplicity and versatility, making it appealing to both rookie and expert developers. Flask's philosophy isfounded on its minimalist approach, giving developers the fundamental tools tobuild web applications while letting them choose their preferred components foradded functionality. Flask was created by Armin Ronacher.

The steps for adding user-generated playlists and real-time chat to your media streaming application using Python Flask

Step 1: Define User-Generated Playlists

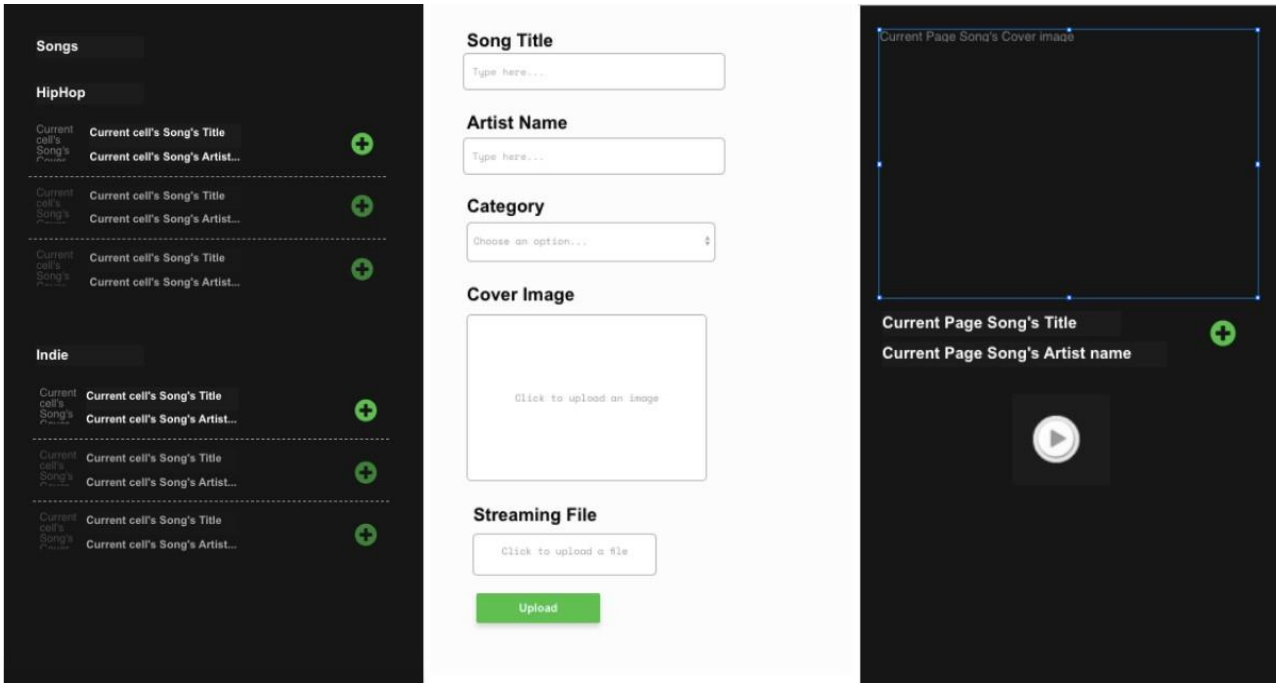
Database Model:

Create a database model for playlists that includes fields like title, description,owner (user ID), and a list of media items (e.g., video or audio IDs) in each playlist.

Step 2: Create User Interface for Playlists

User Interface:

- Design a user-friendly interface where users can create, edit, and manage theirplaylists.
- Allow users to search for and select media content to add to their playlists.


Songs
HipHop
Current cell's Song's Cover
Current cell's Song's Title
Current cell's Song's Artist...
Indie
Song Title
Type here...
Artist Name
Type here...
Category
Choose an option...
Cover Image
Click to upload an image
Streaming File
Click to upload a file
Upload
Current Page Song's Cover image
Current Page Song's Title
Current Page Song's Artist name

Step 3: Implement Routes and Views for Playlists

Flask Routes and Views:

Create Flask routes and views to handle playlist-related actions such as creating,updating, and deleting playlists.

Step 4: Implement User Authentication

User Authentication:

- Implement user authentication to ensure that users can create and manage theirplaylists only after logging into their accounts.
- Verify the owner of each playlist through authentication.

Step 5: Database Operations for Playlists

Database Operations:

Implement logic to add and remove media items from playlists.

Set up Create, Read, Update, Delete (CRUD) operations for playlists in the database.

Step 6: Displaying Playlists

User Interface:

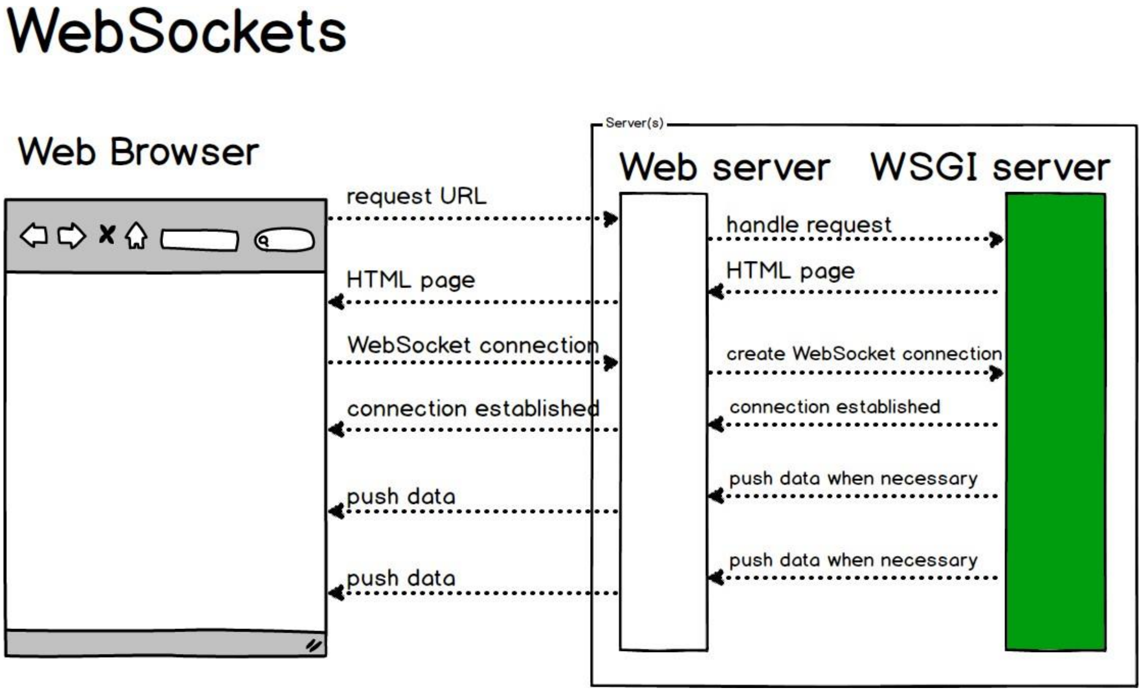
- Design a page where users can view their playlists, including media items ineach playlist.
- Allow users to explore and view the playlists created by other users.

Step 7: Implement Real-Time Chat

Chat Room Design:

- Create chat rooms where users can communicate in real-time while watchingmovies or listening to music.
- Assign a unique identifier to each chat room, and allow users to join these roomsto chat.

Step 8: Use WebSockets for Real-Time Chat



WebSockets and Sockets:

- Implement real-time chat using Flask-SocketIO or Flask-SSE.
- Web Sockets are suitable for enabling instant messaging between users.

Step 9: Design a Chat Interface

Chat Interface:

- Design a user-friendly chat interface within your application where users cansend and receive messages.
- Consider options for displaying user avatars, timestamps, and message history.

Step 10: Enhance Security for Chat

Security:

- Implement security measures to prevent unauthorized access to chat rooms andprotect user privacy.
- Ensure that users can only join chat rooms associated with the media contentthey are currently viewing.

Step 11: Implement User Interaction

User Interaction:

Allow users to join chat rooms related to the media they are currently streaming.Enable users to create, join, leave, and manage chat rooms seamlessly.

Step 12: Moderation and Reporting

Moderation and Reporting:

Implement moderation features to handle inappropriate content and user reports.Allow users to report abusive or spammy messages.

Step 13: Scalability and Compatibility

Scalability and Compatibility:

- Ensure that your chat system can handle a large number of concurrent users.
- Optimize the chat interface for both mobile and desktop users, providing a consistent user experience.

Conclusion:

By following these steps, I can successfully enhance your media streaming application with user-generated playlists and real-time chat. User-generated playlists allow users to create personalized collections of media content, makingtheir experience more engaging and interactive. Real-time chat fosters community and social interaction, enabling users to discuss the content they areenjoying and providing a richer overall experience.