MINI PROJECT – II

(2019-20)

Live Video Streaming Application

MID TERM REPORT

Institute of Engineering & Technology

Chirag Srivastava
Roll No: 171500088

Supervised By

Akash Chaudhary Sir

Department of Computer Engineering & Applications

Abstract

This mid-term Mini Project II report documents the amount of work done in the Mini Project during this date. The report first shall give an overview of the tasks completed during this period with technical details. Then the results obtained shall be analysed. Report shall also elaborate on the future works which are still to be persuaded as an advancement of current work. I have tried my best to keep the report simple yet technically correct. I hope I succeed in my attempt.

Index

Introduction

The major feature of this web application would be the users streaming their videos live, also with an option to watch other user's streams as well. It would list all the streams of other users. The users will have the facility of logging and signing up into their personal accounts with Google's services (OAuth). Then the users after logging into their accounts will have the both options of either creating a new stream or watching other user's streams. In order to stream live the users would need to have OBS (Open Broadcaster Software) installed into their machines so that the screen and other devices like video camera or a microphone could be recorded and integrated into a video feed. Then the stream would be send to the application's server and would be broadcasted online. The application will be developed with all the modern Full Stack technologies. It would be made fully responsive to render across all the different devices. It would also be scalable for future integration and development with production level servers and technologies.

Area of Development

This project deals with the area of Web Development. The whole application runs on React with Redux providing solid state management foundations. It is being developed with major Web Development Technologies like React, Redux, HTML, CSS, JavaScript and Semantic UI.

The Problem Definition

How can we share our ideas or works by broadcasting our screen or our live video to others.

Objective of the project

The purpose of this application is to share the current screen or a live video of the user to a wide audience. And the audience can watch the stream hassle free. In other words to deliver live video streams to other people.

Requirements

a) Hardware:

- A computer or any portable device connected to the internet and with bare minimum specifications.
- Minimum: 1.5Ghz, Dual Core, 2GB of RAM.

b) Software:

- Standard Web Browser like Chrome, Safari or Firefox.
- OBS (Open Broadcaster Software) or any streaming software
- React
- Redux
- NodeJS

Methodology of the project

The project is currently in full development using Web Development technologies. The application uses React for all the developmental work with the power of Redux to handle state management inside the application. The user makes request to either watch a stream or to create one. In case of user watching a stream the request is made from React part of the application to the server which fetches proper video feed. In case of user wanting to create a stream they are first authenticated then they can create, edit or delete those streams associated to their account. In order to stream a user also needs to configure their OBS or any other third party streaming software to stream video feed to our application.

Implementation

Just unlike every other application, this is a perfect implementation of SPA(Single Page Application). Single Page Applications just only renders components that need to be shown to the user, rest everything is abstracted away. The React side of the application are in full compliance of RESTful conventions. The route navigation is made possible by 'React Router' with 'Google OAuth' providing the authentication.

So, if a user wants to stream from his machine they first need to log in to the application. After successfully logging in they can create their stream channel by giving a suitable title and description. Then all they need is to simply send their streams to that particular url or channel key which they get after creating their streaming url. They also have the feature to edit and delete their streams. Now they just need to stream from OBS or any other third party software. Now, when the user streams it is sent directly to the RTMP server (Node Media Server) which communicates with a web server (JSON Server API) containing records of all the streams. Now, if a user wishes to watch a stream they simply send a request to the RTMP server which sends it to the Web Server and finally the web server redirects them to their stream for which they made a request.

Work Progress

The project is almost halfway complete. Most of the backend related work is complete and the rest is under way. At this stage the Web Server API is all set and is responding to our basic requests. The landing page of the application displays the list of available streams to watch. The user can do all the basic operations such as CRUD (Create Read Update Destroy) for the streams. They are properly authenticated before creating, editing or deleting a stream, making sure that they don't edit or delete someone else's streams.

Now, there are things that are still left undone. Most of the styling part is still remaining to implement. The pages are to be styled and made responsive to render across different range of devices. The Google OAuth API is showing some trouble signs when accessed from inside a local network, they only seem to work with a single JavaScript origin defined at the time of creating API credentials. Another major thing that is still remaining is the proper integration of the web server with the RTMP server. The JSON Server works totally fine for prototyping or mocking but something more robust is needed in production level environment.

The application can be cloned or downloaded from:
https://github.com/chiragsrvstv/live-video-streams

References

React Documentation
https://reactjs.org/docs/getting-started.html

Semantic UI Documentation
https://semantic-ui.com/

NPM (Node Package Manager)
https://www.npmjs.com/

Redux
https://redux.js.org/

Axios
https://www.npmjs.com/package/axios

Node Media Server
https://github.com/illuspas/Node-Media-Server

Mozilla Developer's Network
https://developer.mozilla.org/en-US/

JSON Server
https://github.com/typicode/json-server

OBS
https://obsproject.com/

React Router
https://github.com/ReactTraining/react-router