

# AKASHWANI

1<sup>st</sup> Kshitiz Sharma  
Department: Computer Science  
Engineering  
Chandigarh University  
Gharuan, India  
[kshitizsharma.00@gmail.com](mailto:kshitizsharma.00@gmail.com)

2<sup>nd</sup> Akshita  
Department: Computer Science  
Engineering  
Chandigarh University  
Gharuan, India  
[20BCS7342@cuchd.in](mailto:20BCS7342@cuchd.in)

3<sup>rd</sup> Vaibhav Sharma  
Department: Computer  
Science Engineering  
Chandigarh University  
Gharuan, India  
[20BCS7365@cuchd.in](mailto:20BCS7365@cuchd.in)

4<sup>th</sup> Sargun Pal Singh Kohli  
Department: Computer Science  
Engineering  
Chandigarh University Gharuan, India  
[20BCS1515@cuchd.in](mailto:20BCS1515@cuchd.in)

5<sup>th</sup> Isha Chaudhary  
Department: Computer Science  
Engineering  
Chandigarh University Gharuan, India  
[20BCS7342@cuchd.in](mailto:20BCS7342@cuchd.in)

# IJSER

**Abstract—** This is a nodejs based radio which basically helps to transfer information among the group of people and an organization in just one go. It is a mass media communicator which helps to communicate messages among a group of people and also in this era where people have such a hectic schedule that seeing text messages and other important notifications has become tough for them.

**We introduced such a system where they could listen the text message verbally and also any important information does not miss out.**

**Keywords—***node.js*.

## I. INTRODUCTION

AKASHWANI is basically a campus radio which is designed by the students of university for the students of university itself. It basically is a node.js server-based radio which does not require any kind of radio waves or frequencies for the transmission of communications among groups. What all we just need for the transfer of information is all the required information noted at the backend and as

we know that nodejs is one of the emerging technologies these days this will act as both server as well as user interface for the radio . There are a lot of future implementations for the project and also which can be beneficial not only for the university but also for the people in large organizations where information needs to be passed massively. Thus in this era where everyone has a hectic schedule and does not have that much time to see the text messages and in result which leads to loss of some of the important notifications as well we thought to design and implement this idea of nodejs based server radio named as AKASHWANI where things will get easier and messages are transferred easily and not even a single notification remains left out.

A local survey held in our class of computer science and engineering what we found that many people did not have such time that they could see the messages and implement on them and in return many of the important information was left out and they had to suffer from various losses such as placement drives, some important tests such as class assignments, co-curricular activities and many more .

So it stuck our mind that yes, there should be such an application which could help people in reaching out each and every important information so that suffering is less and none of them gets affected whether it be placement point of view or any other thing. So this AKASHWANI basically will serve as a platform to communicate among a large group of people and deliver information easily in just one go. It has been proved that verbally listening is more powerful than visualization so we have decided to make this prototype which could be beneficial for each and every organization.

It basically bridges the gap between organizations and universities as each and every information will be announced massively so that each and every person comes to know about the happenings in the universities. So what we conclude is that not only it provides a good medium for the transferring of information but it also reduces gap among various organizations and communities as they get inter-connected with this medium .

The prototype consists of four parts mainly:

**Playlist:** Where we will have all the songs of the songs, announcements, and other necessary information to be given.

**Queue:** Queue will consist of all the ready to be played queued songs.

**Now Playing:** This section will be showcasing the song playing at the current moment.

**Controls:** All the controls assigned to the keys will be included in this section.

The control box will contain the hardcoded list of keyboard key bindings.

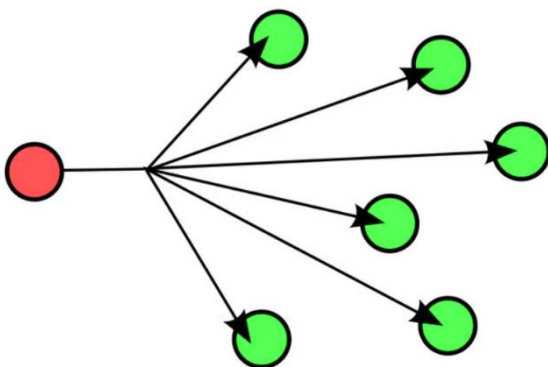


Figure:1

Node.js helps for one-to-many transmission. Therefore, it has been used for streaming the information from one end to multi-end.

One-one communication basically means transferring information from one end that is there is no concept of

feedback in it. Information once sent is sent and this cannot be reverted back in one-one communication.

Mass communication means to transfer information among the large group of people in just one go, which is also the definition of broadcasting.

## II. RELATED WORK

As a part of literature survey, we investigated some applications of campus radios and already existing works that already exist in market. The aim is to observe how these applications work and to see how they can be improved and how are they different. To date it is identified that there are some radios which work under frequencies and radio waves but any such application based radio which does not use any such radio waves and frequencies have not been identified yet.

Educational radios have been there in existence since there was broadcasting. In 1925 there were approximately 170+ universities and organizations that have been working under the concept of educational radios but that was basically based on radio frequencies and radio waves.

The unavailability of any such radio where one could transfer information within an organization without using radio waves was one of the ideas behind making AKASHWANI.

Radio has the ability to transfer information massively and it works on one-one communication.

There have been various radios, applications in the university that have been transferring information among students and group of people but any such application or radio that transfers the information verbally among the group of people has not yet been invented yet. So an idea to design such an application where we could transfer information verbally among a large group of people without any risk of loss of data stuck our mind.

As people having a lot of hectic schedules don't have time to see the text messages and other documentations and as per the research verbal listening is more effective than written documentations this nodejs server based radio would be a good project to work upon.

We just have designed the prototype which consists of features such as playlist,queues,now playing and control box for navigation of keys. Though it is just a research

outcome and is under progress we have various future scopes for the project which can be implemented in future. Functionality of mic, schedule of shows to be played are some of the functionalities which could be added in future scope of the project.

Thus, this project has a very vast and good future ahead.

### III. PROPOSED WORK

In the proposed work we have just designed a prototype which broadcasts the messages using node.js server and has playlist at the backend. As discussed in the literature review there have been many radios in the history which work on traditional technologies such as radio waves, frequencies and many more. But as there was unavailability of any such media which could transfer the information in just one go to a group of people without the use of frequency waves and other traditional technologies such as radio waves. We came across node.js which basically is **an open-source development platform for executing JavaScript code server-side**.

Node is useful where we need to develop applications that our server side and is often used for real time applications such as chats, broadcasting, web push notifications and many more. The main goal of the proposed prototype is to implement a Node.js HTTP server that will have an endpoint intended for streaming songs, i.e., audio data. A client could then start consuming the data by making a request to that endpoint. (I)

The proposed system also consists of the control panel which consists of the management of songs that need to be streamed out.

The starting top will consist of a play button which will play all the songs which will be similar to that of online radios available.

There are four main sections that are necessary for the simple but convenient control panel. Those are:

1. Playlist – list of all available songs
2. Queue – list of queued-up and ready-to-be-played songs
3. Now Playing – name of the song that is streaming at the moment
4. Controls – keyboard key bindings instructions.

The controls box will contain hardcoded list of keyboard key bindings.

The now Playing box in the proposed system will just have one item: the song that is currently being played.

The Playlist box will be used for storing the list of all the songs from the directory where the app has been run.

Queue is the last box we need to implement. Similar to the Playlist.

Stream Magic is the API used to send the song to every consumer that is listening at that particular moment. Thus, the main use of stream magic is to stream and broadcast the

songs from user end to a large group of people in just one go.

Stream Magic is used for broadcasting purpose.

In, our project we need to have a one-to-many (broadcast) transfer of data, and we need the producer to be in charge of the speed of the transfer in a push-based manner. The reason for this is quite obvious: we want all of the consumers to receive approximately the same part of a song at a single point in time.

### ADVANTAGES

- Beneficial for students as well as organizations who want to transfer information in just one go, and without taking the risk of loss of information. As it has been proved that verbal listening is more impactful as compared to text messages and documentations.
- Messages can be transferred at one go that means we can send messages to the large group of people by broadcasting. Broadcasting basically means sending messages to the group of people in just one click.
- All the things happening in the surrounding can reach people easily as this can also help in ease of transfer of information.
- It can also help in creating great connections with the community members as it will connect people of different communities together with the help of broadcasting.
- AKASHWANI is easy to set-up and also there is no need to for any such frequencies and radio waves required for it as it is node.js server based Radio.
- It will also overcome the disadvantage of traditional radios which basically focussed on radio waves and bandwidth as it will be using node.js as a server and won't be requiring any such kind of bandwidths for transferring information among the group of people.

FEATURES OF THE PROJECT

This is a nodejs. Server based project which will tranfer the information among group of people without the use of bandwidth, radio waves and other methods which the traditional radios were using.

Because of the hectic schedule of people there came a need of such a technology which could help people get information easily and effectively so that none of them gets missed.

This consists of main four sections that are necessary for the simple but convenient control panel. Which basically are:

- 1.Playlist: Playlist will contain of all the available songs.
- 2.Queue: List of all queued-up and ready-to-be played songs will be there in the queue section.
- 3.Now Playing: Name of the song which is currently playing will be displayed in this section.
- 4.Controls: Keyboard key binding instructions. The control box will contain hardcoded list of keyboard key bindings.

The front-end of the webpage will be displayed with a play and pause button which will be streaming all the songs. The future scope of the project is to design a mic-functionality for the radio for more effective use of radio.

This is the class diagram of AKASHWANI which will be giving us a blueprint of how our radio will be designed.

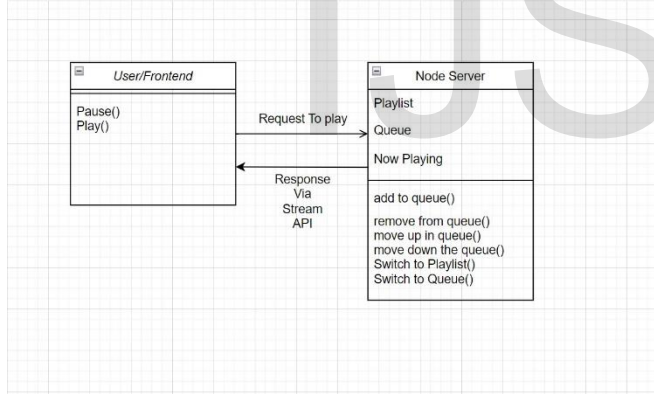


Figure:2

IV IMPLEMENTATION

How to get started

In order to launch the application, we first need to execute the server code written on the machine it will then allow us to run a custom command named ‘node-radio-mini’ in terminal which would then allow us to open any music file



Figure:3

- Open Any Music Directory with mp3 files.
- Run node-radio-mini command on terminal
- A playlist as shown below will be displayed

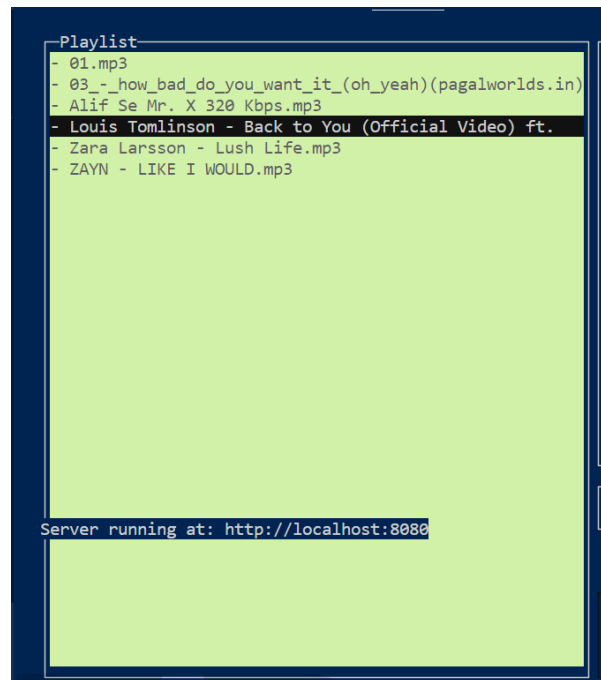


Figure:4

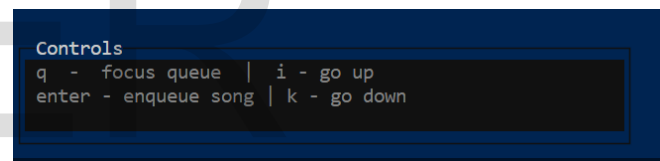


Figure:5

Use the controls to enqueue the songs of playlist

The Localhost 8080 will emit our server output.

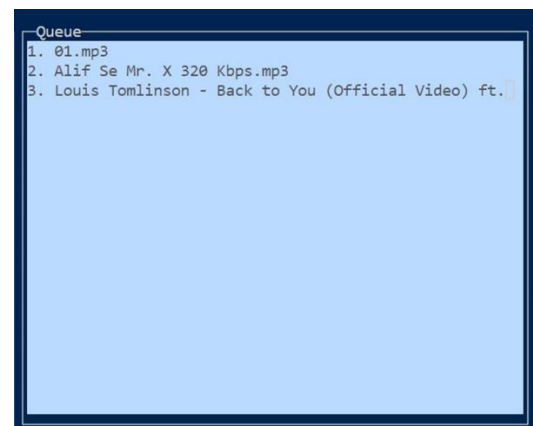


Figure:6

The Queue will be as shown above.

All the songs which we have added from the playlist in the queue to be played are displayed here.

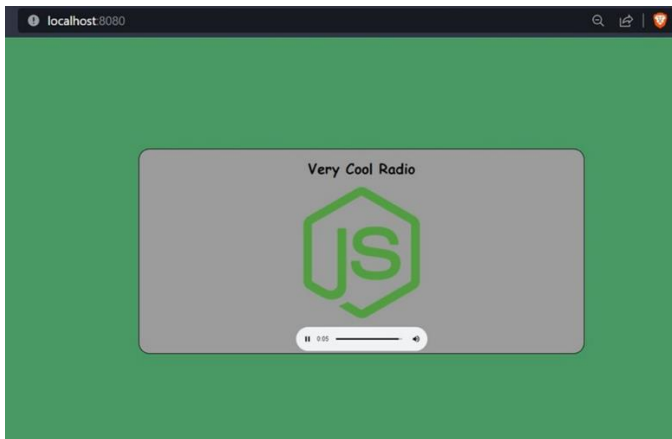


Figure:7

This will be the final outcome on the user end an application that plays live radio from our back-end server.

#### APPLICATIONS AND OUTCOMES

- 1.To transmit information. E.g.: - Any information related to fests, events both technical and cultural or any other information related to university can be shared over our radio website.
2. To transmit educational content. e.g.: - We would be updating students about the company visits in the campus, No. of successful placements, interview tips and live podcasts with students who cracked big MNC's.
3. To transmit entertainment content. E.g.: - We would be providing platform to the students who are good at singing, poetry or storytelling. In this way the users would find the platform much more interactive and students with talent would also get a stage.

#### FUTURE SCOPE

##### **Mic functionality:**

We can add mic functionality in it so that we could announce all the important notifications to the users listening the radio.

##### **Text announcements:**

In future we will add the functionality of announcing texts.

##### **Upcoming section:**

In future an upcoming section can also be added which will show all the upcoming events in the radio channel.

##### **Most Listened and Top Rated:**

A section which could tell u the most listened and top-rated shows can be added to improve the experience of users listening to the radio.

#### CONCLUSION

We have developed a Server based Radio which does not require any kind of radio waves or frequencies for the transmission of information. We can easily broadcast information over the internet using the AKASHWANI Radio. It reduces the gap among various organizations and communities as they get interconnected with this medium. It will also overcome the disadvantage of traditional radios which basically focused on radio waves and bandwidth as it will be using node.js as a server and won't be requiring any such kind of bandwidths for transferring information among the group of people. It can be a most efficient way to broadcast the information as due to the hectic schedule of people there is a need of such technology where user can just open web app and can get the information rather than opening traditional FM radios and then finding radio signal which becomes hectic in the busy schedule and can get information easily and effectively so that none of them miss the information. Thus, this is just a prototype which has a great future ahead.

#### REFERENCES

- 1) [HTTPS://WWW.MONGODB.COM/LANGUAGES/MERN-STACK-TUTORIAL](https://www.mongodb.com/languages/mern-stack-tutorial)
- 2) [HTTPS://WWW.YOUTUBE.COM/WATCH?V=YCEZEWXUN4](https://www.youtube.com/watch?v=YCEZEWXUN4)
- 3) [HTTPS://WWW.GEEKSFORGEEKS.ORG/HOW-TO-PLAY-PAUSE-VIDEO-OR-AUDIO-IN-REACTJS/](https://www.geeksforgeeks.org/how-to-play-pause-video-or-audio-in-reactjs/)
- 4) [HTTPS://UXDESIGN.CC/UI-CHEAT-SHEET-RADIO-BUTTONS-CHECKBOXES-AND-OTHER-SELECTORS-BF56777AD59E?GI=F1E00ABBE22F](https://uxdesign.cc/ui-cheat-sheet-radio-buttons-checkboxes-and-other-selectors-bf56777ad59e?gi=f1e00abbe22f)
- 5) [HTTPS://WWW.YOUTUBE.COM/WATCH?V=W-9TIB1NPJ4](https://www.youtube.com/watch?v=W-9TIB1NPJ4)

IJSER