

Targeted Digital Signage Network

Project Report as a part of "Internet of Things" course



**Semester II
M.Tech (I year)
IIIT Bangalore
7th May 2015**

**By,
Team **Show-offs**
Pratesh Kumar Reddy. R (MT2014519)
Santhosh K V (MT2014103)
Satish Kulakarni (MT2014106)**

Contents

Introduction:.....	3
Motivation behind the project:.....	3
System Model.....	4
Basic Overview.....	4
Hardware Components Used.....	5
Software Components Used.....	5
Communication used.....	5
Description of functionalities at Node (Screen) side :.....	5
File structure inside each Node (screen).....	5
Working Of Each Node:.....	6
Description of Server functions and WebApp:.....	6
Overview:.....	6
Implementation:.....	7
Use Cases:.....	7
XML files:.....	7
Image and Video Files:.....	8
Services:.....	8
Navigating the webApp:.....	8
Screenshots:.....	9
Future Work:.....	14
THANK YOU.....	14

Introduction:

The project is aimed at developing a 'Targeted Digital Signage Network'. Digital Signage is a method of displaying the content like digital images, video, streaming media, live websites over Digital displays like LCD, LED and Projectors. In college campuses it can be used to display the upcoming events/talks, various announcements, welcome note to the guests.

So the goal is to

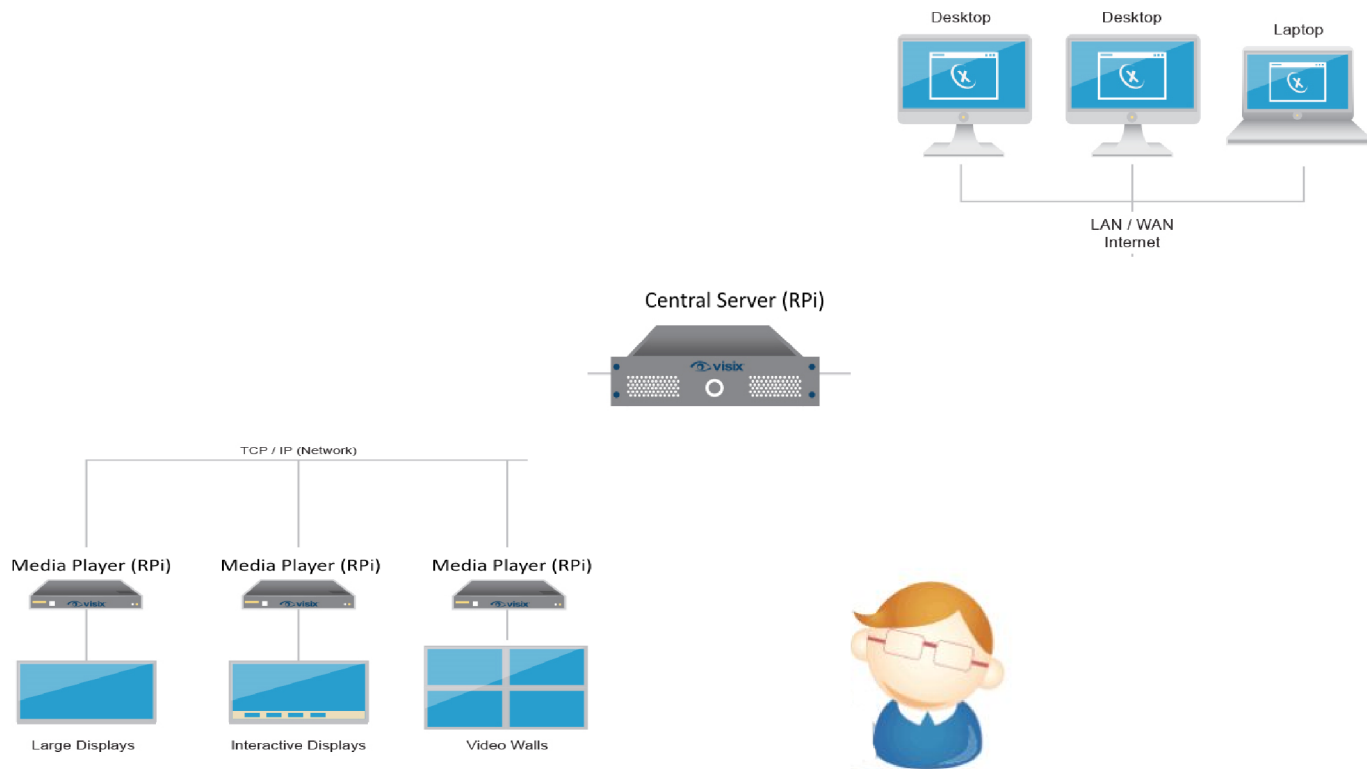
- Create a network of digital displays (nodes) that dynamically loop through the contents and display them.
- Connect the displays to a central server that hosts the entire data.
- Create a web-browser based graphical user interface that can be used to manage the content to be displayed at the nodes.
- Authenticate the content managers and enable them to add/delete/modify the content from any remote PC.
- Make it 'targeted' by identifying the users and display the content that is more relevant to that category of users.

Motivation behind the project:

Digital Signage is latest trend because it allows monitoring the large number ad-displays or information-displays from a single control centre. This area has been made more affordable by introduction of cheaper decent-performance computing platforms like 'Raspberry -pi'. Our main motivation was to use these cheaper computing platforms to enable a cost-effective Digital Signage Network (DSN) in a relatively controlled environment like university campus. With this DSN deployed in the campus it will be easy to **broadcast the information** on important events going on in the campus. As can be seen later, the extra features developed in this DSN will make it very much suitable for campus information broadcast.

System Model

Basic Overview



Node side:

Two Raspberry-Pi boards will be used that will be connected to two HDMI LCD/LED displays (nodes). Each of these node R-Pis will display the contents in their set of local folders in a continuous loop. So these nodes will be displaying videos, images, websites etc., in a full screen mode under continuous loop using video player, image viewer and browser respectively.

These node R-Pis will be connected to the central server through internet and thus can regularly synchronize the content from the server. Each of the node R-Pi will synchronize the data from the central server and store it in local folders.

The node R-Pi s will also have a RFID reader connected to them that can be used to detect the user (who carries a RFID tag). The reason to choose Raspberry-Pi is its ability to even display HD videos, connect to internet and its low cost.

Server side:

We will maintain an apache based server in another computer that will have the master copy of contents to be displayed in each node. At present we are hosting the sever locally. The server will host a web-page that contains the GUI to login into the server and then to upload/delete/edit the content in the server. These changes in the server folders will eventually be reflected in node R-Pi's folders.

Hardware Components Used

- One Linux PC to host the HTTP and FILE server
- Two Raspberry Pis to act as end-nodes.
- Two displays (LCD) with HDMI ports to display the event content
- One EM-18 RFID reader (3cm range)
- Three 125khz RFID tags

Software Components Used

- Python 2.7 for handling the functionalities at node end
- Java
- Struts2
- JSP(Java server pages)
- MySQL(database)
- Apache Tomcat 7(Http Server)
- Apache 2(File Server)

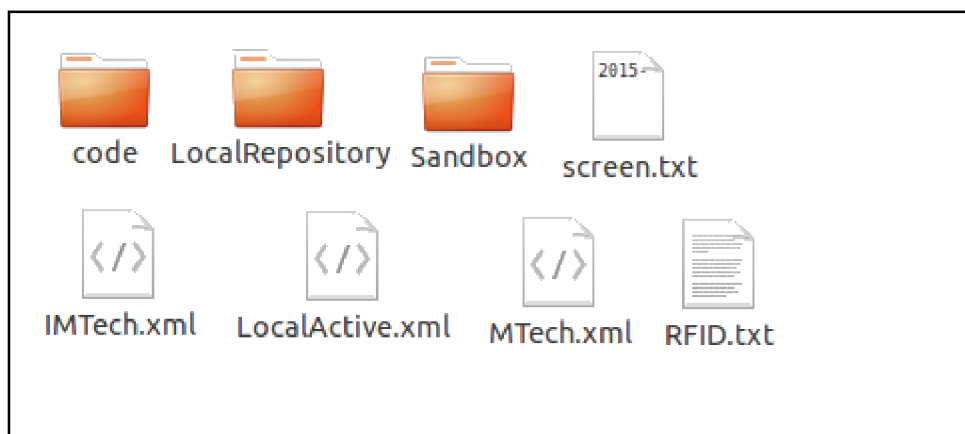
Communication used

1. The node R-Pis will connect to Server using internet (at present the server is hosted locally. So we connect to server using the server's local ip address).

Description of functionalities at Node (Screen) side :

We have used two nodes in this project. Each node has one R-Pi and one LCD display and each R-Pi is connected to the internet.

File structure inside each Node (screen)



CODE : This folder will contain the python codes that are entirely responsible for creating the events and display in the screens.

LOCALREPOSITORY: This folder will contain the actual videos/images that are to be displayed. It does not contain the text events because those events are directly written into .xml files

SANDBOX: This file will hold temporary files that are to be downloaded (from server) before actually modifying the other main files in the node.

SCREEN.TXT: This text file will contain the last date and time when any event has been added/deleted/modified in this code.

RFID.TXT : This file will contain the name of RFID tag (if any) that is placed near the RFID reader.

LOCALACTIVE.XML: This file will contain the attributes of all the events to be shown all the times.
IMTECH.XML : This file will contain the attributes of all the events to be shown whenever the IMTECH student (with IMTECH RFID tag) is near the RFID reader.

MTECH.XML : This file will contain the attributes of all the events to be shown whenever the MTECH student (with MTECH RFID tag) is near the RFID reader.

Working Of Each Node:

The node end takes care of the following functionalities:

1. Download the temp.text file from server and compare with time in local screen.txt to decide if there were any changes.
2. If there are any changes then download the LOCALACTIVE.XML, IMTECH.XML, MTECH.XML from the server into SANDBOX folder of node.
3. Download any new files that we uploaded into server to LOCALREPOSITORY of this particular node.
4. Delete the files from LOCALREPOSITORY that were deleted in server.
5. Then copy the LOCALACTIVE.XML, IMTECH.XML, MTECH.XML into main path of node to replace the respective old files. Now the changes at server side get replicated in at the node also.
6. Now by parsing the .XML files and using the files in LOCALREPOSITORY, the NewWebPageCreator.py will create the webpages. This webpages will be launched in a web-browser in full-screen mode. These webpages will redirect automatically from one to the next webpage. This effectively looks like the events are looping continuously.
- 7.



NewWebPageCreator.py



Downloader.py

The functions from 1 to 5 are implemented by Downloader.py code while function 6 is implemented by NewWebPageCreator.py code.

Description of Server functions and WebApp:

Overview:

This application provides an interface for the user to add/modify/delete various events that are to be displayed in the Digital Signage Network. The application provides options for the administrator to assign the time, date and also the time period for the which the event has to be displayed (All available options are explained in following sections). He can also select to whom the event is to be associated with(Mtech, iMtech or All).

The WebApp for our project was done using the J2EE platform. We have used the following technologies for our WebApp development:

- Java
- Struts2
- JSP(Java server pages)
- MySQL(database)
- Apache Tomcat 7(Http Server)

- Apache 2(File Server)

Implementation:

Use Cases:

Following are the various use cases we have considered:

1. Add new event (file or text event).
2. Delete an existing event.
3. Edit existing events.

Login is common to all of these use cases and the actor is only the administrator. Multiple persons can login at a time from different terminals and modify/upload/delete the events in the website (or webapp).

XML files:

We are using xml files for making the data parsing easier. Following are xml files that are used in the project:

global.xml

Path: /var/www/html/dsnproject/GlobalRepository/global.xml

This is the global file which has all the information required at the raspberry pi to display general information that the administrator has uploaded. Only those whose status is active are to be displayed. So this xml file is generated using only those events for which the status in the database is "active".

Localactive.xml

Path:/var/www/html/dsnproject/Screen_number/ Localactive.xml

Screen_number = Screen1 or Screen2

This file holds all the details for only the corresponding screen (Screen1 or Screen2). Each raspberry pie will be downloading only the files that are needed for the screen it is associated with. Parsing this xml file, the raspberry pi will be able to display all the details that are associated with the corresponding screen.

iMtech.xml

Path:/var/www/html/dsnproject/Screen_number/iMtech.xml

This file holds all the iMtech file details for only the corresponding screen (Screen1 or Screen2). Parsing this xml file, the raspberry pi will be able to display all the details that is associated with the corresponding screen for having label as "iMTech".

Mtech.xml

Path:/var/www/html/dsnproject/Screen_number/

This file holds all the Mtech file details for only the corresponding screen (Screen1 or Screen2). Parsing this xml file, the raspberry pi will be able to display all the details that is associated with the corresponding screen for having label as “MTech”.

The Class CreateXMLFile in the org.iiitb.dsn.xml will have all the methods required to get all the details from the database and creating these xml files accordingly. Whenever these XML files are generated, we are also updating a temp.txt file in each of the Screen_number folders to indicate that there is a change in the xml file. The raspberry pi checks these files and decides whether to download xml files or not.

Image and Video Files:

We are storing all the image and video files locally in the raspberry pi as well as in the server. All the files which are added/updated, will have a tag ID and will be uploaded to a common location called the GlobalRepository in the path /var/www/html/dsnproject/GlobalRepository. All the files will be downloaded by the raspberry pi from this location and will then be stored in its local repository.

Services:

The project uses a service called RemoveOutdatedEntries to change the status of those events whose end date & time has crossed. This service runs every fifteen minutes and after change of status is done, all the xml files are also updated and new files are downloaded by the raspberry pi.

Navigating the webApp:

Whenever adding a new event, we have two options of either to add a text event or image/video event.

For adding it will ask for various attributes, almost all the attributes are self-explanatory.

LOOPTIME attribute is the amount of time a particular event has to stay stationary.

PRIORITY is “how important an event is?” So a high priority event will loop more frequently than a medium priority event and a medium priority event will loop more frequently than a low priority event.

STATUS: Only the ‘active’ events will be shown at the nodes. The inactive events will not be shown until they are made active (even if their starting time has reached). And the events which have crossed their end time will be automatically made inactive by the server.

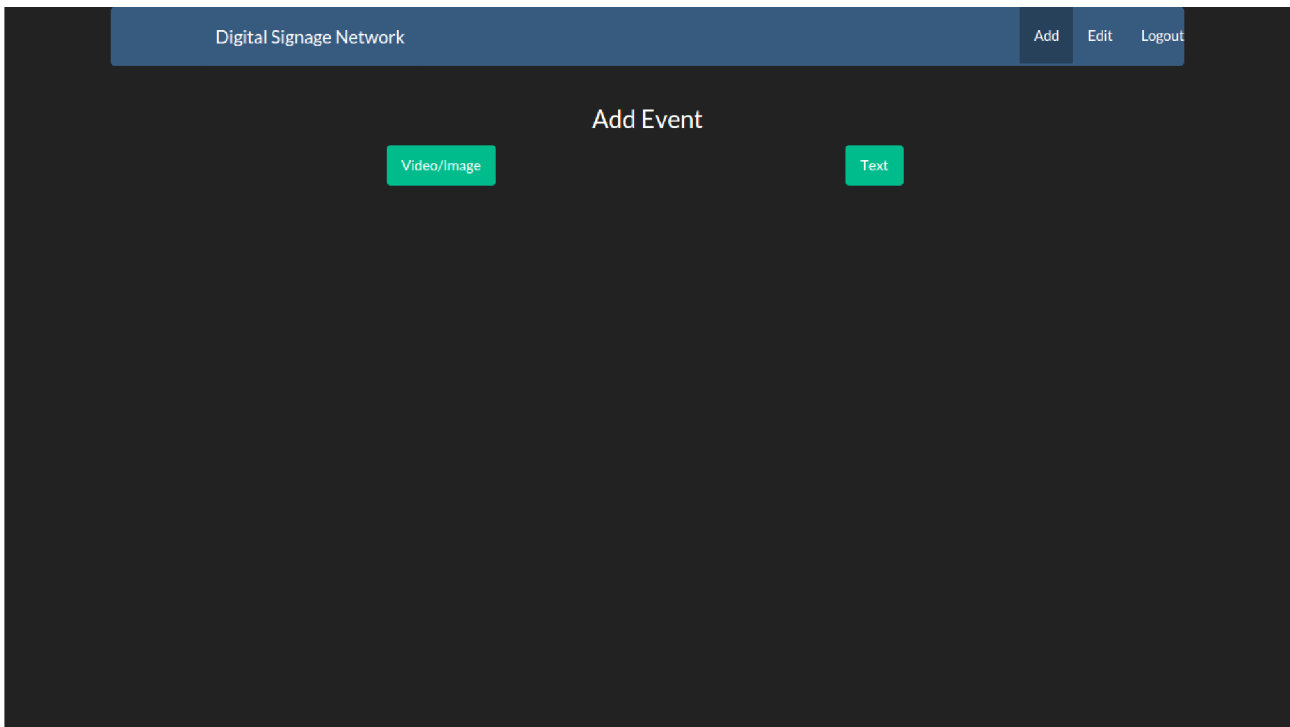
Making an event as inactive will only make a particular event not to be shown in the displays but it will not delete any file/text related to the event either in the server or in the any nodes. But deleting a particular event will entirely delete the event/text from all the node (screens) and server.

Screenshots:

Following are the screenshots and their corresponding descriptions:



Fig 1.1: Homepage to enter username and password



Fig

2.1: Page to choose among text event or file to be uploaded

The screenshot shows the 'Upload File' page within the 'Digital Signage Network' application. The header is identical to the previous page. The main content area has a dark background with the title 'Upload File' centered at the top. Below the title, there is a form with several fields: 'Topic' (text input with 'NewFileForScreenshot'), 'Screens' (a list box with '1' and '2' selected), 'File' (a text input with a 'Choose File' button and 'library.jpg'), 'From Date' (date input with '05/07/2015'), 'To Date' (date input with '05/08/2015'), 'From Time' (time input with '10:00 AM'), 'To Time' (time input with '03:00 PM'), 'Loop Time' (text input with '3'), 'Label' (dropdown menu with 'MTech'), 'Priority' (dropdown menu with 'High'), and 'Status' (dropdown menu with 'Active'). At the bottom of the form is a 'submit' button.

Screen for adding image/video event and corresponding details.

Add Text Event

Topic:

Screens:

1

2

Text to Display:

This is for the screenshot in the report

From Date:

To Date:

From Time:

To Time:

Loop Time:

Label:

MTech

▼

Priority:

High

▼

Status:

Active

▼

Screen for adding a new Text Event with all the corresponding details.

Digital Signage Network
Add Edit Logout

Edit File

Topic:

Screen:

1

2

File:

Choose File

im32000

From Date:

04/30/2015

To Date:

05/03/2015

From Time:

06:00 AM

To Time:

09:00 PM

Loop Time:

Label:

MTech

▼

Priority:

Medium

▼

Status:

Inactive

▼

! Please select an item in the list.

Select validation if none of the screens are chosen.

Digital Signage Network						Add	Edit	Logout
Edit Files								
Topic	Screens	Status	Priority	Related To	Edit	Delete		
Less Important Notice	1	Inactive	Low	All				
Important Notice	1,2	Inactive	Medium	MTech				
Summer semester Fee	1,2	Inactive	High	MTech				
Welcome to IIITB	1,2	Inactive	Medium	All				
Today's Quote	1,2	Inactive	Low	All				
IMtech exam time table	1,2	Inactive	High	iMtech				
Sports for iMtech	1,2	Inactive	High	iMtech				
Strike today	1,2	Inactive	High	All				

localhost:8080/iotNew/editTextAction?editId=68

Editpage: This page gives a list of all the events currently in the system and required details. An option for editing or deleting the events is provided.

Digital Signage Network				Add	Edit	Logout
Edit File						
Topic:	<input type="text" value="Sports for iMtech"/>					
Screen:	<div>1</div> <div>2</div>					
File:	<input type="button" value="Choose File"/> No file chosen			72_TT.JPG		
From Date:	<input type="text" value="mm/dd/yyyy"/>	04/30/2015				
To Date:	<input type="text" value="mm/dd/yyyy"/>	03/03/2015				
From Time:	<input type="text" value="--:--"/>	06:00 AM				
To Time:	<input type="text" value="--:--"/>	09:00 PM				
Loop Time:	<input type="text" value="4"/>					
Label:	<input type="text" value="MTech"/>					
Priority:	<input type="text" value="High"/>					
Status:	<input type="text" value="Active"/>					
<input type="button" value="submit"/>						

EditFileEvent page: Here the user can change all the attributes of a particular image/video event.

Edit Text Event

Topic:

Screens:

Text to Display:

From Date: 04/30/2015

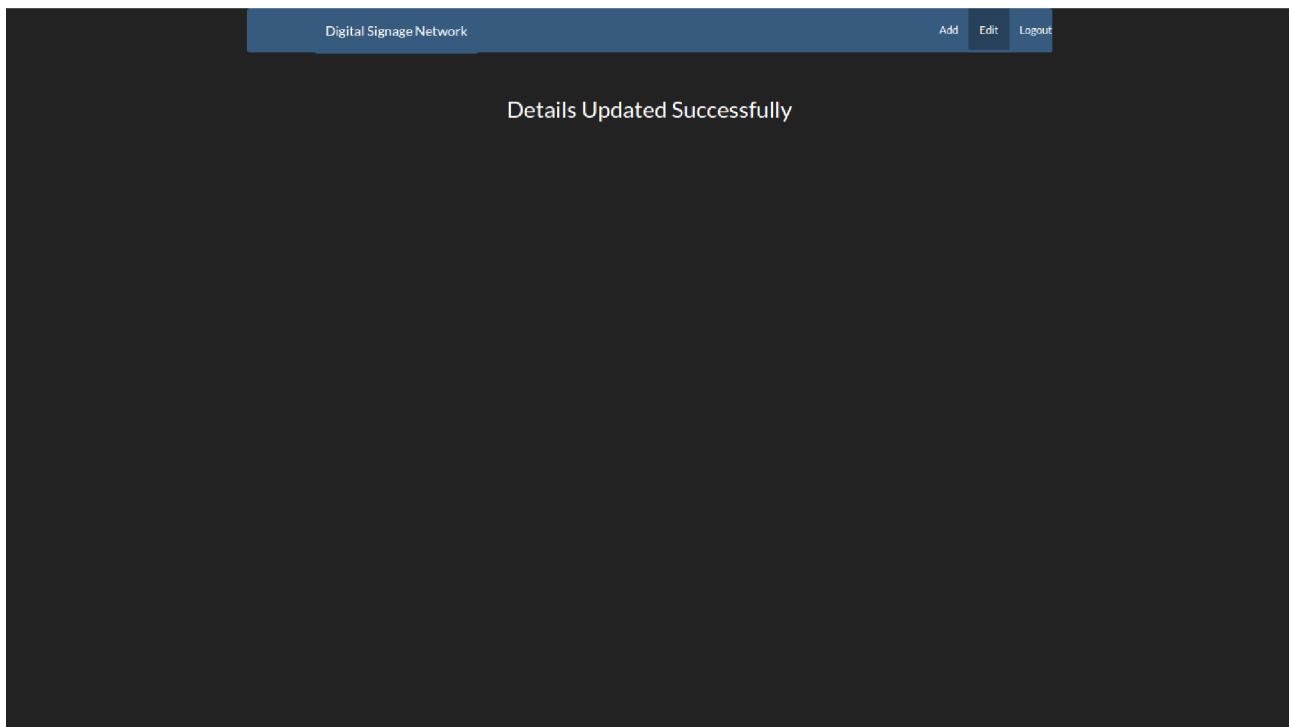
To Date: 05/05/2015

From Time: 02:00 AM








To Time: 09:00 PM

Loop Time:

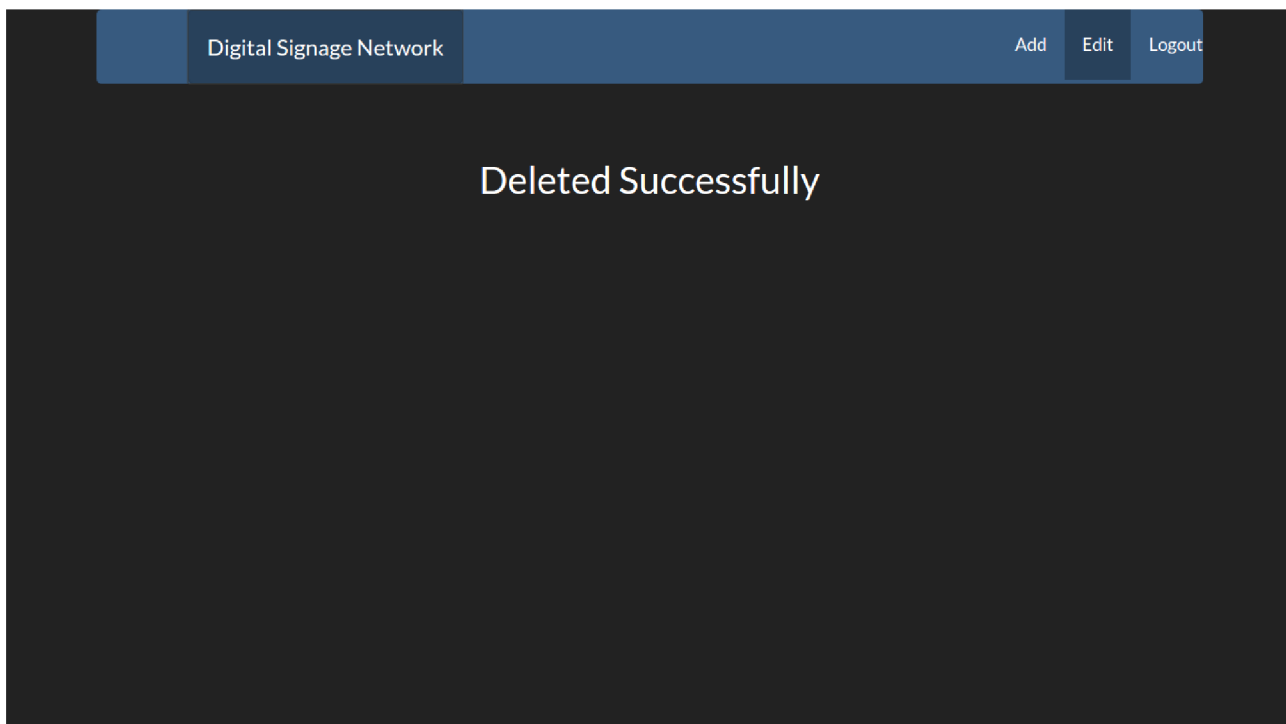
EditTextEvent page: Here the user can change all the attributes of a particular text event.



Update Successful message

Edit Files						
Topic	Screens	Status	Priority	Related To	Edit	Delete
Less Important Notice	1	Inactive	Low	All		<input checked="" type="checkbox"/>
Important Notice	1,2	Inactive	Medium	MTech		<input checked="" type="checkbox"/>
Summer semester Fee	2	Active	High	iMtech		<input type="checkbox"/>
Welcome to IIITB	1,2	Inactive	Medium	All		<input type="checkbox"/>
Today's Quote	1,2	Inactive	Low	All		<input type="checkbox"/>
iMtech exam time table	1,2	Inactive	High	iMtech		<input type="checkbox"/>
Sports for iMtech	2	Inactive	Medium	MTech		<input type="checkbox"/>

Deletion of various events can be done simultaneously. Check the delete option for the event you want to delete and press the “Delete Selected” button.



Delete Successful message

Various validations required for the proper operation of the WebApp have been done so that the user will be able to upload the correct data and the system works perfectly. All the updating, deletion of files, downloading, uploading of files are done automatically and gives a good user experience.

Future Work:

To improve the GUI such that it can show summary of each event and it can retrieve back the previous info of an already present event whenever the user tries to edit it.

To make it more robust and secure so as to use it effectively in our college campus.
To make it more intelligent such that even if some send our server an email describing an event, it should be able to parse the mail, know the dates of the event and also main content of the event and add it to the list of events on its own.

THANK YOU