ITMD 465/565 Assignment 2 “IIT News”

Spring 2015 – 14% course grade

Objectives

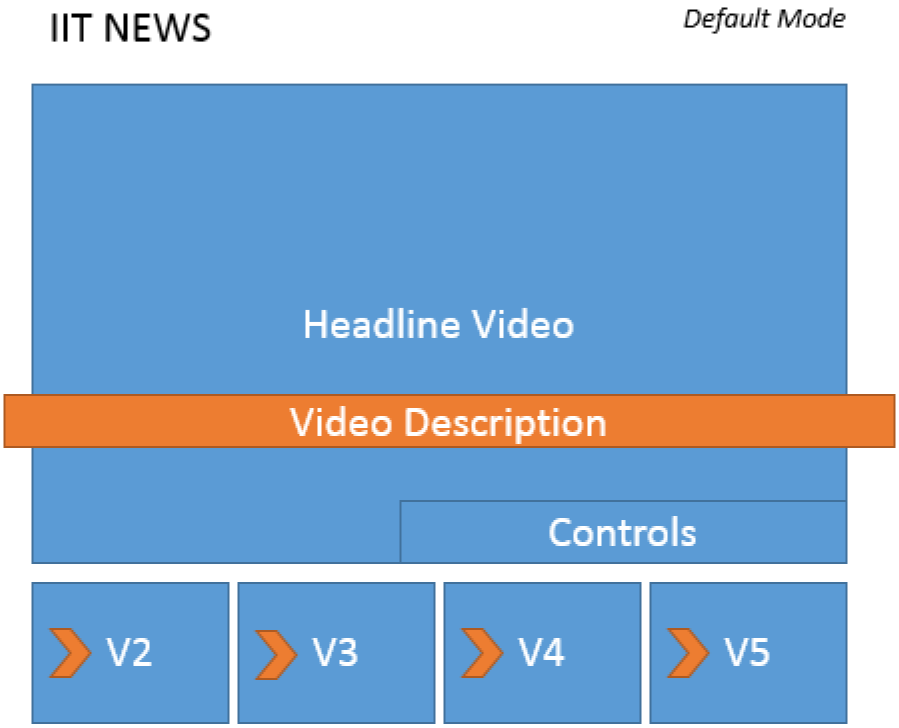
- Use traditional and assisted AJAX techniques for asynchronously loading content
- Compare and contrast the use of jQuery animations and CSS3 animation features.
- Describe and use responsive design principles and techniques and explain the advantages of responsive design as it relates to desktop and mobile web-based applications.
- Explain the functionality and develop applications that use the HTML5 Local Storage API for making internet applications available when not connected to the internet
- Describe the use of the HTML5 Audio and Video APIs and compare and contrast the HTML5 APIs to legacy audio and video solution

Instructions

- You are to modify the included **assignment2.html** to provide the following functionality:

IIT News would like to design a rich new cross platform video application – you are designing the first version of such a cross platform application by creating a JavaScript library that will create a custom video player that will load a selection of movies and enable the user to select a movie to play. Videos will be displayed from the day’s top headlines, and users should be able to pick one of the videos they have not scene before to view it – and additional information about the video should become visible as the video plays. IIT News wants to compete on the ease of use of its video player and has decided it must have the following features:

- Your application must show usage of the jQuery library
- The list of videos should be obtained from downloading and parsing the supplied AJAX file
- It should arrange the videos in random order on the page according to the general layout guideline displayed below
    - Each video should display the thumbnail and the main video must also display the title
- The video description should appear on the video after the video has completed by 25% or more.
- The video description should use a css3 animation to slide in from the left
- You must store in localStorage what videos the user has watched on their browser and display those at the end of the list (displaying unwatched videos first)
- You must implement a complete custom set of video player controls – including:
    - Play/Pause Button
    - Progress Bar
- The video’s should first display a thumbnail of the videos, using the supplied URL in the json file, which – when clicked – will start the video playing.

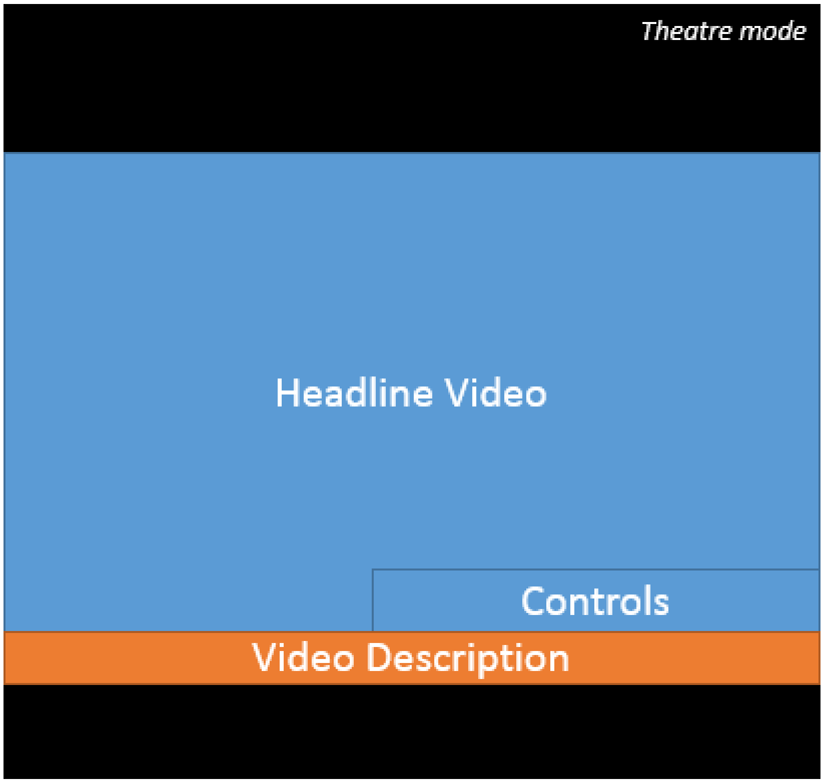


Steps

Graduate Extension

In addition to completing the undergraduate requirements:

- You must also utilize an additional control to create a “theatre” mode in which the lower videos disappear and the video description is instead shown (after 25% of the video has elapsed) at the bottom of the video. This mode must take up the full screen of the browser – the transition to theatre mode should be smoothed via jquery, css3 transitions or animations. Use the layout guideline below to help.
- When in this mode you must detect when the video has finished and display the next video
- You must also store the progress of the current video in localstorage when paused, and attempt to re-seek to its saved progress if the page is refreshed or otherwise loaded again.



Graded

To obtain maximum grades:

- Build all functionality entirely in JavaScript
- Attach all elements to the included hook on the page
- Have no JavaScript errors in console during operation
- Create a readme file explaining your submission

Submission

Put all of your code, and any additional files you create into a zip file named: **FIRSTNAME_LASTNAME_ANO_A2.zip** and submit that on blackboard by the due date – Saturday 11:59PM Chicago Time - March 28th 2014.

Late Policy

You will receive a 10% grade reduction per 24 hours your submission is late, starting from 1 minute after the deadline. **Last minute submissions lead to last minute problems – start and submit early! Extensions are only given for documented extreme hardship and considered before 48 hours prior to the deadline.**

Data

Data is supplied in the included clips.json file – which should be loaded via AJAX.

Web Server

Included with the submission files is a Node.js web server – Node.js is a technology that we will use in more depth later in the course – but for now, it will work as our web server to serve html/json files.

It is a standalone application that can run on most operating systems. Make sure to download the correct installation for your operating system.

To start the application – use either run.bat (windows) or run.sh (linux/mac). You can alternatively run it yourself by using the commands:

```
cd server

(linux/mac) node server.js <desired port number if other than 8888>

(linux/mac) node.exe server.js <desired port number if other than 8888>
```

You should then see some output like the following:

```
node.exe server.js

Static file server running at

  => http://localhost:8888/

CTRL + C to shutdown
```

The server will then stay operating until you kill this process. All files starting from the ../www/(path) directory will be available through your browser on http://localhost:888/(path)

Place your html/js/json files within the www/ directory. When you submit your code for this assignment, submit only your code within the www directory, do not submit the server directory – we will use our own node.js installation – this will save on the upload/download size.

Data File

Also included with this assignment is a datafile: clips.json – use this file to load data for your application.