

Last updated: 2015.02.24

CSCI 4140 – Tutorial 6

Assignment 2 Overview (Client Side)

Hints on Front-end Development

Matt YIU, Man Tung ([mtyiucse](mailto:mtyiucse@cse.cuhk.edu.hk))

SHB 118

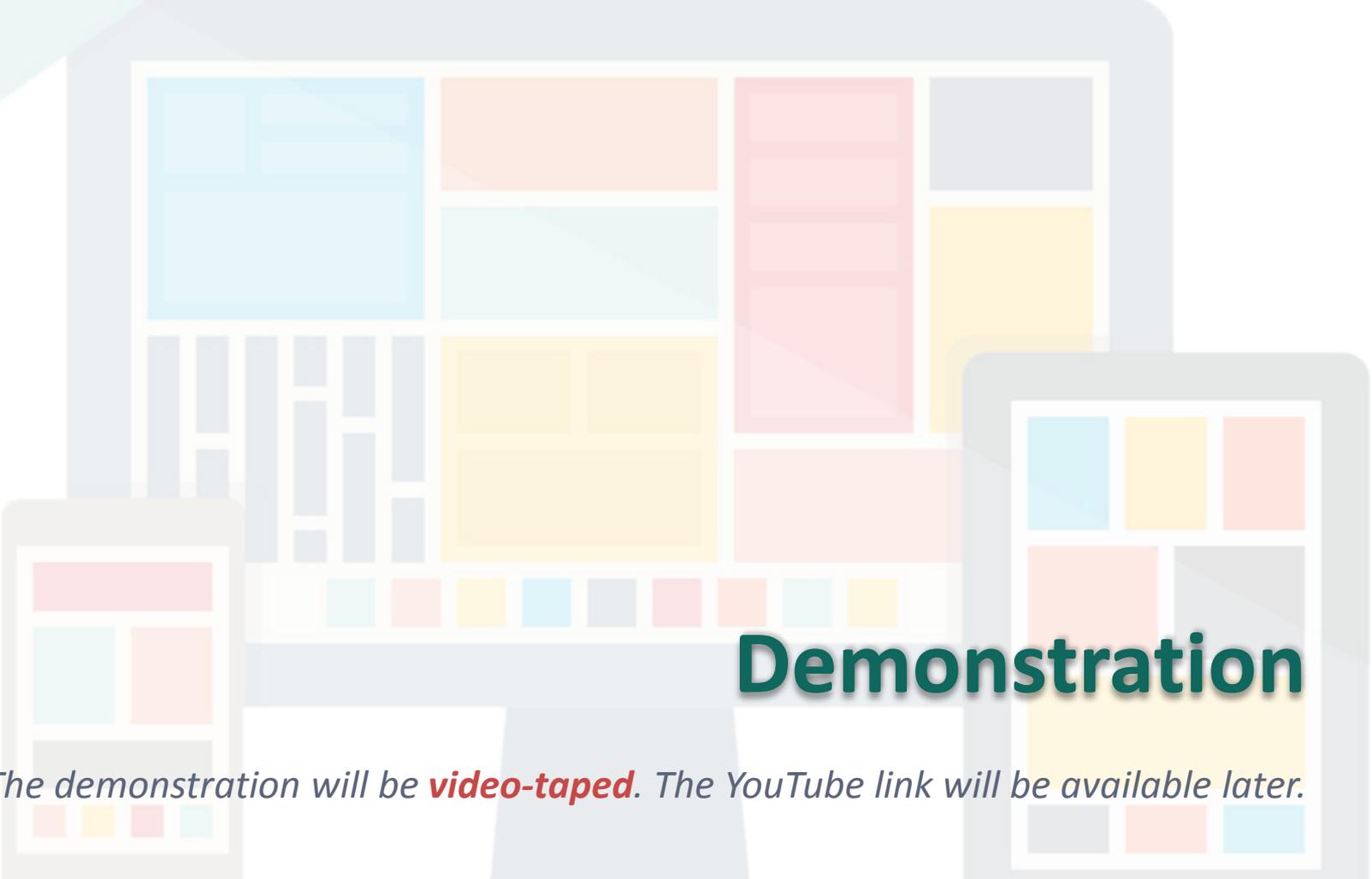
Office Hour: Tuesday, 3-5 pm

2015.02.26

Outline

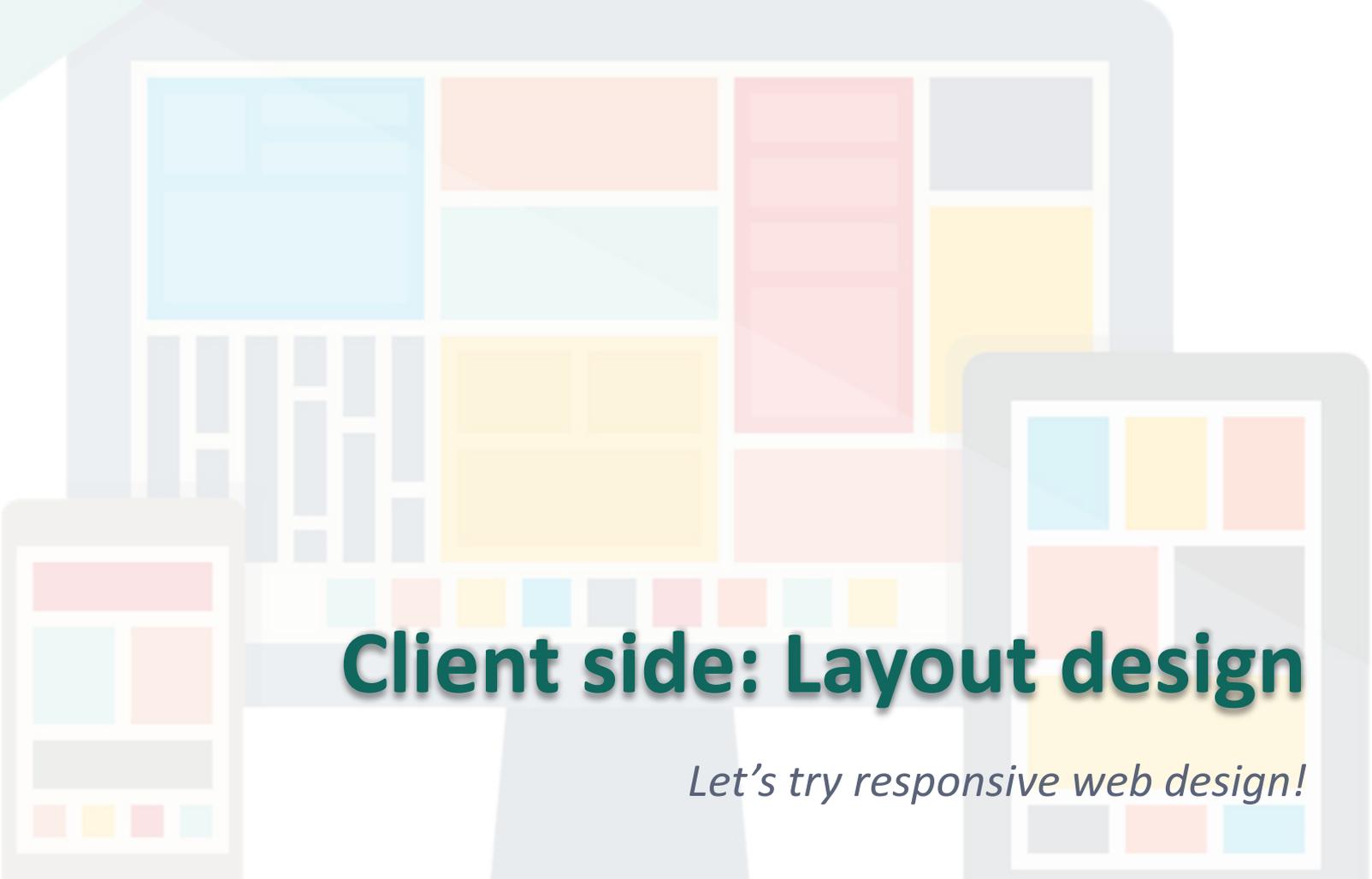
- Demonstration
- Client side
 - Layout design
 - QR code display
 - Playlist management
 - YouTube player control
- ***Next tutorial***
 - Work flow
 - Server side





Demonstration

*The demonstration will be **video-taped**. The YouTube link will be available later.*



The background features a stylized illustration of a laptop and a tablet. The laptop screen shows a complex wireframe layout with various colored rectangular blocks (blue, orange, red, yellow, grey) representing different content areas. The tablet screen shows a simplified version of the same layout, demonstrating responsive design. The text 'Client side: Layout design' is overlaid on the center of the image.

Client side: Layout design

Let's try responsive web design!

CSCI 4140: Assignment 2

When screen size ≥ 992 px...

YouTube Remote (Desktop)

YouTube Player

CSCI4140 Ruby on Rails for Dummies

CSCI4140 Guest Lecture Series
Ruby on Rails for Dummies

2015.02.09

CC BY NC SA
This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License <http://creativecommons.org/licenses/by-nc-sa/4.0/>

YouTube Remote Control

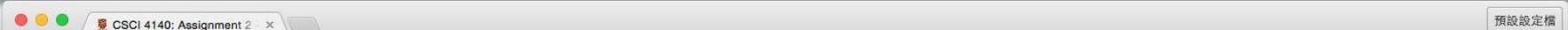
Playlist

- M7lc1UVf-VE: YouTube Developers Live: Embedded Web Player Customization
- sQANr6r4Kms: CSCI4140 Ruby on Rails for Dummies

Enter Video ID or URL +

CLEAR ALL

asgn2-desktop.png



When screen size ≥ 992 px...

CSCI 4140: Assignment 2



YouTube Remote (Desktop)

A screenshot of a YouTube player interface. The video title is 'CSCI4140 Ruby on Rails for Dummies'. The video thumbnail features the 'RAILS' logo. The video description includes 'CSCI4140 Guest Lecture Series Ruby on Rails for Dummies' and the date '2015.02.09'. A Creative Commons license logo (CC BY-NC-SA) is visible at the bottom of the video player.

YouTube player
(More details in **this** tutorial)

A screenshot of a 'YouTube Remote Control' panel. It features a row of three buttons: a green play button, a red pause button, and an orange stop button. Below these are two larger buttons: a purple left arrow and a blue right arrow. At the bottom is an orange bar with a volume icon and a notification bell icon.

Control panel

A screenshot of a 'Playlist' interface. It shows a list of videos with titles like 'M7lc1UVf-VE: YouTube Developers Live: Embedded Web Player Customization' and 'sQANr6r4K... CSCI4140 Ruby on Rails for Dummies'. Each video entry has a close button (X). Below the list is a text input field labeled 'Enter Video ID or URL' with a plus button and a 'CLEAR ALL' button.

Playlist

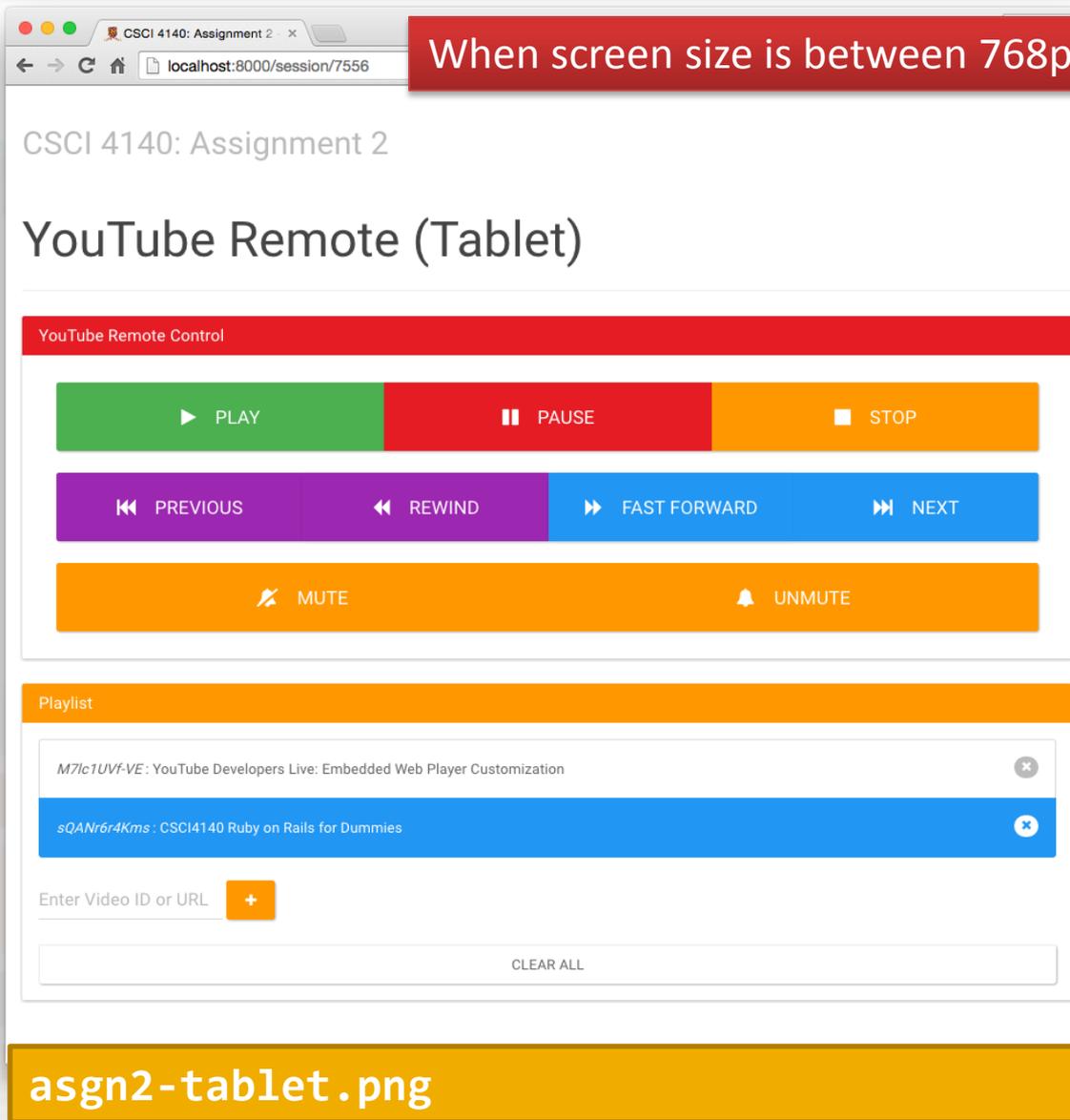
asgn2-desktop.png

When screen size ≥ 992 px...

The buttons are displayed with icon only (WITHOUT description).

asgn2-desktop.png

When screen size is between 768px and 992 px...



CSCI 4140: Assignment 2

YouTube Remote (Tablet)

YouTube Remote Control

▶ PLAY ⏸ PAUSE ■ STOP

⏮ PREVIOUS ⏪ REWIND ⏩ FAST FORWARD ⏭ NEXT

🔇 MUTE 🔊 UNMUTE

Playlist

- M7lc1UVF-VE: YouTube Developers Live: Embedded Web Player Customization
- sQANr6r4Kms: CSCI4140 Ruby on Rails for Dummies

Enter Video ID or URL +

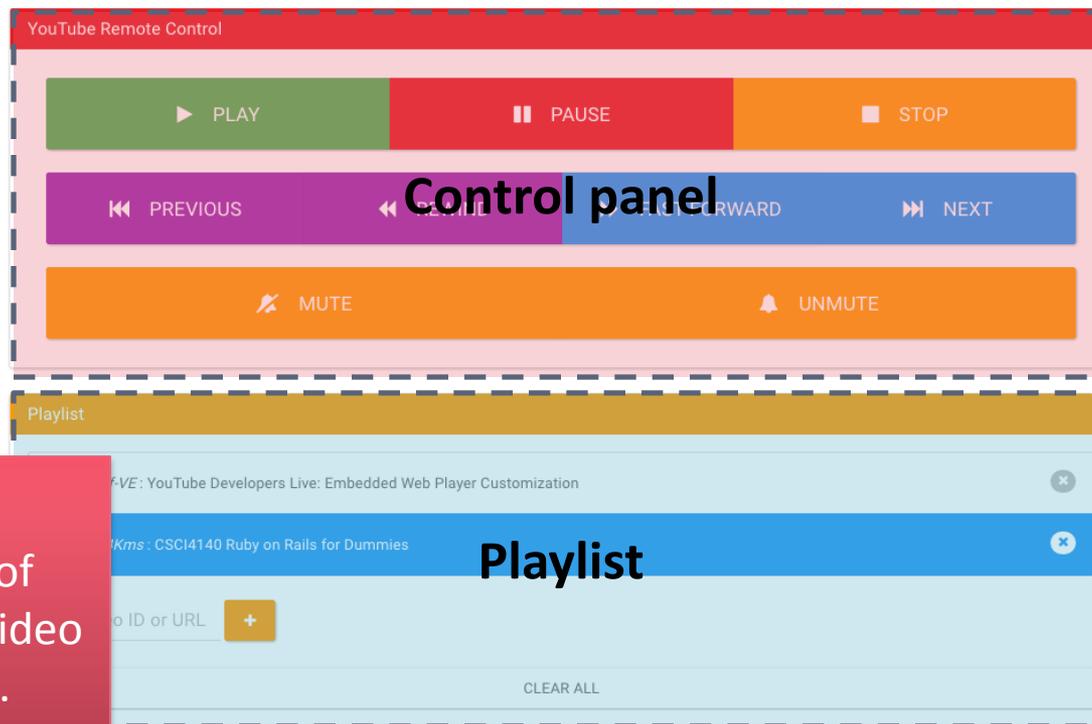
CLEAR ALL

asgn2-tablet.png

When screen size is between 768px and 992 px...

CSCI 4140: Assignment 2

YouTube Remote (Tablet)



Note: When we reduce the width of the window, the video player is removed.

asgn2-tablet.png

When screen size is between 768px and 992 px...

The buttons are displayed with both icon and description.

CSCI 4140: Assignment 2

YouTube Remote (Tablet)

YouTube Remote Control

▶ PLAY ⏸ PAUSE ■ STOP

⏮ PREVIOUS ⏪ REWIND ⏩ FAST FORWARD ⏭ NEXT

🔇 MUTE 🔊 UNMUTE

Playlist

- M7lc1UVF-VE: YouTube Developers Live: Embedded Web Player Customization
- sQANr6r4Kms: CSCI4140 Ruby on Rails for Dummies

Enter Video ID or URL +

CLEAR ALL

asgn2-tablet.png

When screen size is < 768px...

CSCI 4140: Assignment 2

YouTube Remote (Mobile)

YouTube Remote Control

▶ || ■

⏮ ⏪ ⏩ ⏭

🔊 🔔

Playlist

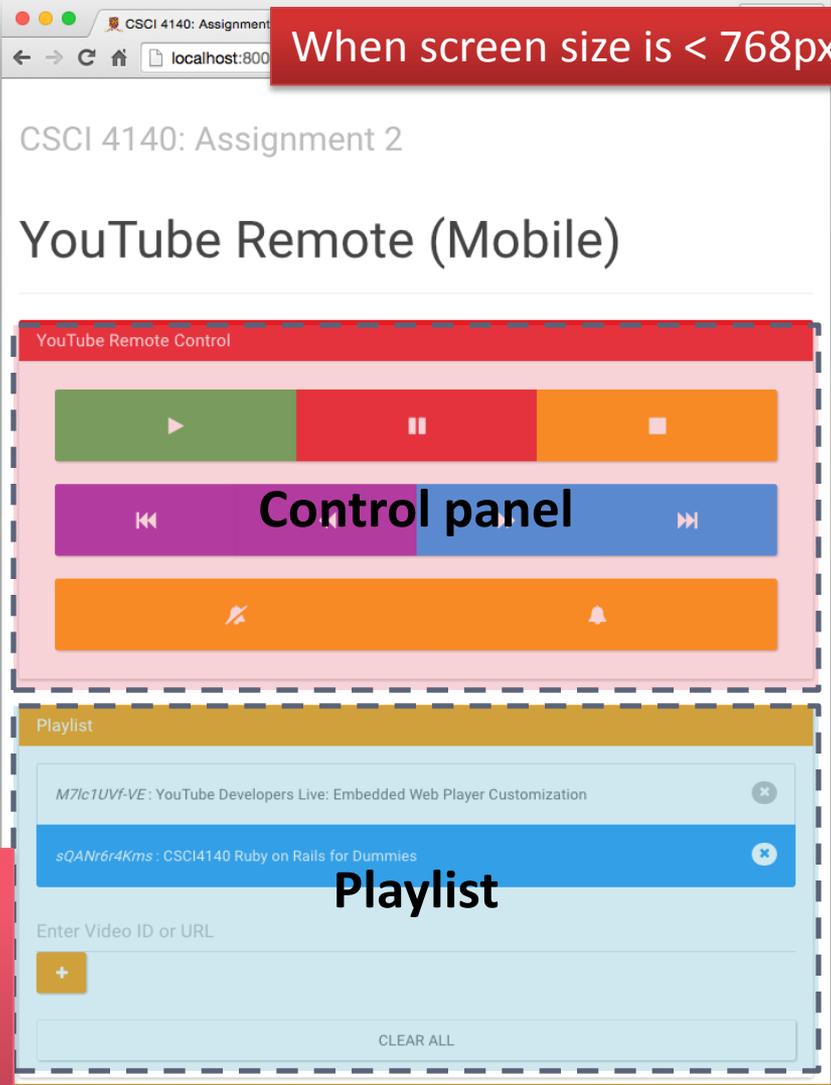
- M7lc1UVf-VE: YouTube Developers Live: Embedded Web Player Customization
- sQANr6r4Kms: CSCI4140 Ruby on Rails for Dummies**

Enter Video ID or URL

+

CLEAR ALL

asgn2-phone.png



When screen size is $< 768\text{px}$...

CSCI 4140: Assignment 2

YouTube Remote (Mobile)

YouTube Remote Control

Control panel

Playlist

asgn2-phone.png

The screenshot shows a mobile browser window with the URL localhost:8000. The page title is 'CSCI 4140: Assignment 2'. The main heading is 'YouTube Remote (Mobile)'. Below the heading, there is a 'YouTube Remote Control' section with a red header. This section contains three rows of buttons: a green play button, a red pause button, and an orange stop button; a purple previous button and a blue next button; and an orange volume icon and a bell icon. Below this is a 'Playlist' section with a yellow header. It contains two video entries: 'M7lc1UVf-VE: YouTube Developers Live: Embedded Web Player Customization' and 'sQANr6r4Kms: CSCI4140 Ruby on Rails for Dummies'. There is an input field for 'Enter Video ID or URL' with a plus button and a 'CLEAR ALL' button.

Note: When we reduce the width of the window, the video player is removed.

When screen size is < 768px...

CSCI 4140: Assignment 2

YouTube Remote (Mobile)

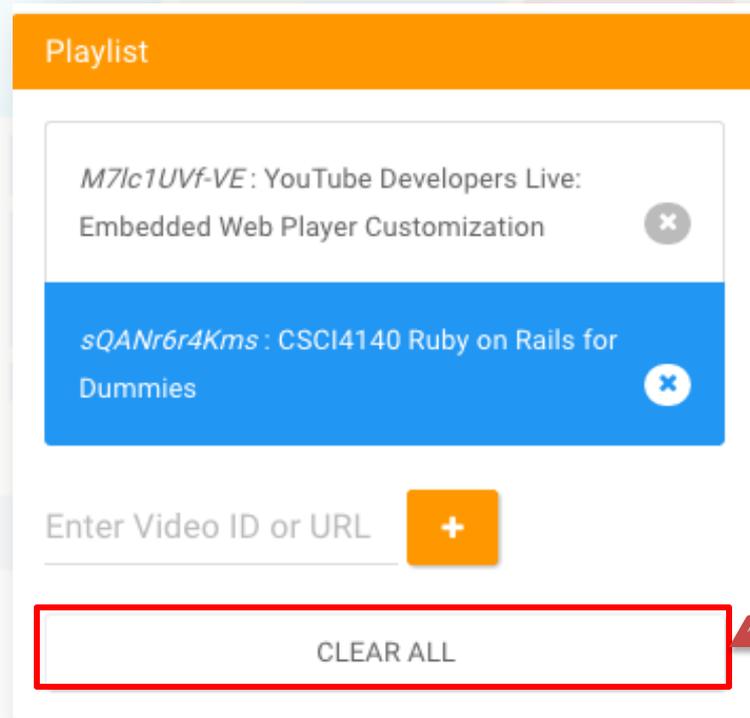
Remote Control

The buttons are displayed with icon only (WITHOUT description).

asgn2-phone.png

Client side: Layout design

- If you already implemented the layout by following the screen capture from last tutorial slides, please add a “**Clear All**” button for the playlist:



Sorry...this button is missing in last tutorial slides... Please add it back.

Client side: Layout design

- You are asked to implement a **responsive UI**
- We will resize the window **WITHOUT refreshing the page**, so you are forced to do the screen width detection in **client side**
 - Suggested solution: Bootstrap / CSS media queries
 - Not recommended: JavaScript
- You are free to rearrange the components, but they must meet the requirements in the specification
- Read the slides from last tutorial if you didn't come

Note: If you want to clone the UI of my site, you can download the high quality screen capture from the resource page 😊

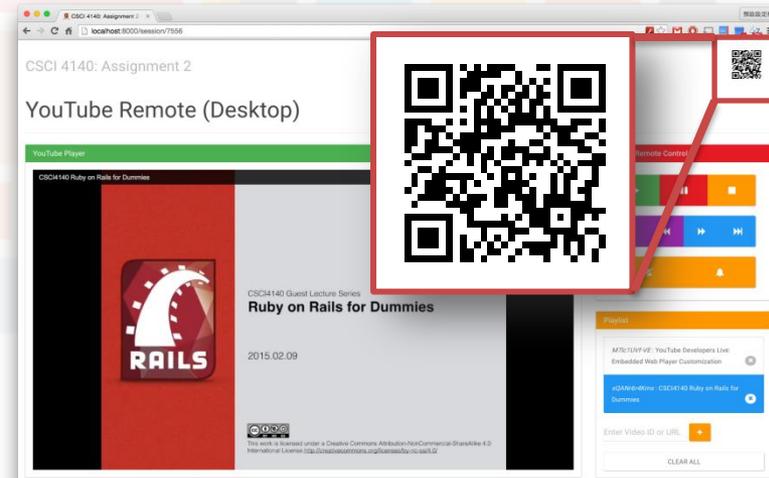


Client side: QR code display

Updated

Client side: QR code display

- You need to display a **QR code** on the page
- The QR code should only contain the URL of the current page
- You can generate the QR code on server side or client side
- Use “**location.href**” to get the URL of the current page in JavaScript on **client side**
- Use **Google Chart** to generate QR code:
 - https://developers.google.com/chart/infographics/docs/qr_codes



Client side: QR code display

- Google Chart Example:

```
<html>
<head><title>QR Code Demo</title></head>
<body>
  
</body>
</html>
```

qr_code/qr_code.html



Updated

Client side: QR code display

- Google Chart Example:

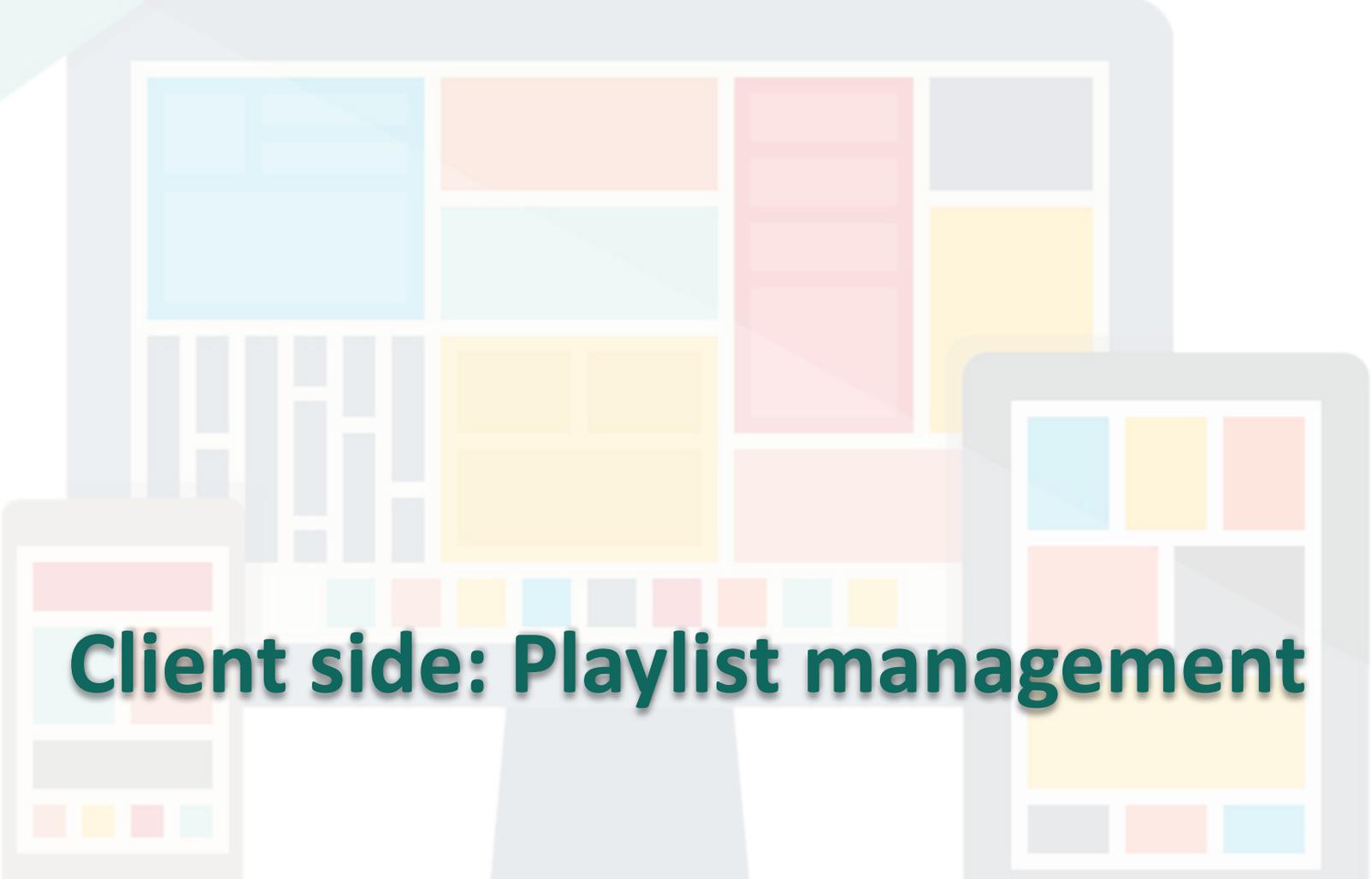
```
<html>
<head><title>QR Code Demo</title>
<body>
  
</body>
</html>
```

Specify a QR code.

Specify the image size.(cht=<width>x<height>)

The data to encode. If you need to specify a URL, it should be UTF-8 URL-encoded. The JavaScript `encodeURIComponent()` function can do so. (Ref.: [\[MDN\]](#))

- How to combine it with “location.href”?
 - Hint:** Generate the `` tag in JavaScript (e.g. by `document.write`)

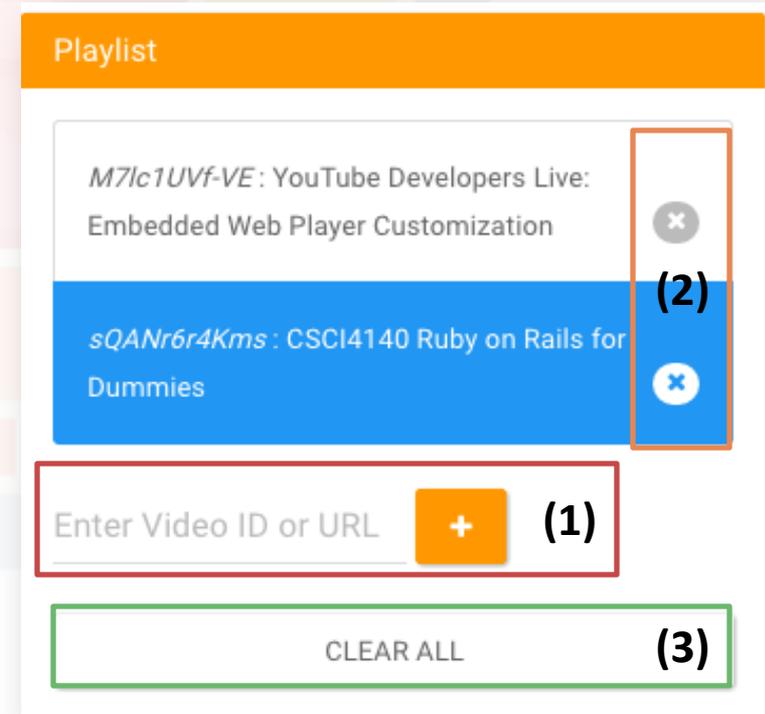


Client side: Playlist management

Client side: Playlist management

- You need to manage a playlist in the application
- Operations supported:
 - Add new video (with its video ID) to the playlist **(1)**
 - Remove a existing video from the playlist **(2)**
 - Clear all videos in the playlist **(3)**
- Of course, the playlist should be displayed in your web page
- Suggested format:

`<Video ID>: <Video title>`



Client side: Playlist management

- When a video is added to or removed from the playlist, the change should be reflected in the UI
 - Use **DOM scripting**
 - Useful functions:
 - `document.getElementById()`
 - `document.querySelector()` (in HTML5)
 - `createElement()`
 - `appendChild()`
 - `removeChild()`
 - Read the lecture notes: “JavaScript (Part 1)”

Updated

Client side: Playlist management

- You don't need to store the playlist on the server
- Storing it on client side is sufficient
 - But, let's forget about cookies
- Use HTML5 **localStorage** (*much simpler!*):
 - To get an item with a key:

```
var i = localStorage.getItem( <key> );
```
 - To set an item with a key

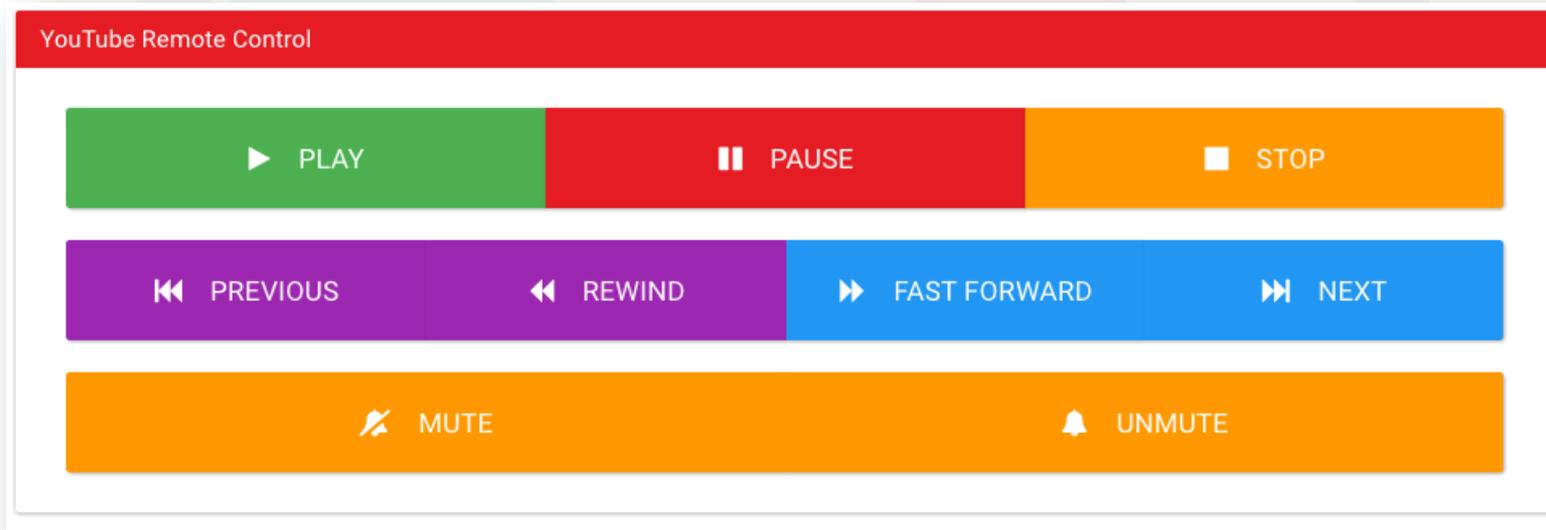
```
localStorage.setItem( <key>, <item> );
```
 - Note: The data type of the item should be **String**
 - Use "**JSON.stringify()**" to convert an array / object to a String



Client side: YouTube player control

Client side: YouTube player control

- You need to implement the following functions with **YouTube IFrame Player API**



- Read the corresponding tutorial slides for more details

Updated

Reminder

- You should finish the **UI design**
- You should start implementing the **player** and **control logic**
- Implement all **playback control functions** which controls the video player on the same page
 - **Hint:** Wrap all YouTube IFrame API calls in your functions. This is useful for extending to support remote control
- **Coming weeks:** Assignment 2 Overview (Server Side) & Back-end development
 - We will use **Node.js**, **Express** and **Socket.IO** on server-side
 - Remember to install Node.js and Express on your computer

– End –