

CSCI 4140 – Tutorial 7

Assignment 2 Overview (Client Side)

Hints on Front-end Development

Matt YIU, Man Tung ([mtyiucse](mailto:mtyiucse@gmail.com))

SHB 118

Office Hour: Wednesday, 3-5 pm

2016.03.10

Outline

- Demonstration
- Client side
 - Layout design
 - QR code display
 - Playlist management
 - YouTube player control



A stylized illustration of a laptop and two tablets. The laptop screen is filled with various colored rectangular blocks (blue, orange, red, yellow, grey) arranged in a grid-like pattern. The two tablets in front of the laptop also display similar colorful block patterns. The background is a light green and white geometric design.

Demonstration

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

An illustration showing a large laptop screen, a tablet, and a smartphone, all displaying a grid-based layout design. The design consists of various colored rectangles (blue, orange, red, yellow, grey) representing different content blocks. The layout is shown in a way that demonstrates how it would adapt to different screen sizes, with the text 'Client side: Layout design' overlaid on the laptop screen.

Client side: Layout design

Let's try responsive web design!

When screen size ≥ 992 px...CSCI 4140: Assignment 2
YouTube Remote

YouTube Player

2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

Task 3: Link Multiple Docker Containers

Command List

- docker run with the --link flag

Instructions

We now have two containers: one web server and one MySQL database. We want to build a web application that has the following architecture:

```
graph LR; Client((Client)) -- Request --> WebServer[Apache Web Server]; WebServer <--> MySQL[(MySQL Database)];
```

Type 'help,' or '^n' for help. Type '^C' to clear the current input statement.

```
mysql> USE hello_world
Database changed
mysql>
```

YouTube Remote Control

⏮

⏪

⏩

⏭

⏮

⏭

Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)

Enter Video ID or URL

+

Clear All

asgn2-desktop.png

When screen size ≥ 992 px...CSCI 4140: Assignment 2
YouTube Remote

YouTube Player

2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

Task 3: Link Multiple Docker Containers

Command List

- docker run with the --link flag

Instructions

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

Client → Request → Apache Web Server ↔ MySQL Database

YouTube Remote Control

Control panel

Playlist

- n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)
- h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2 (Previous version)

Enter Video ID or URL

Clear All

asgn2-desktop.png

CSCI 4140: Assignment 2
YouTube RemoteWhen screen size ≥ 992 px...

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

YouTube Player

2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

Task 3: Link Multiple Docker Containers

Command List

- docker run with the --link flag

Instructions

We now have two containers: one web server and one MySQL database. We want to build a web application that has the following architecture:

```
graph LR; Client((Client)) -- Request --> WebServer[Apache Web Server]; WebServer <--> MySQL[(MySQL Database)];
```

mysql> USE hello_world
Database changed
mysql>

YouTube Remote Control

Play, Pause, Stop, Previous, Next, Full Screen, Volume

Playlist

- n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)
- h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)

Enter Video ID or URL +

Clear All

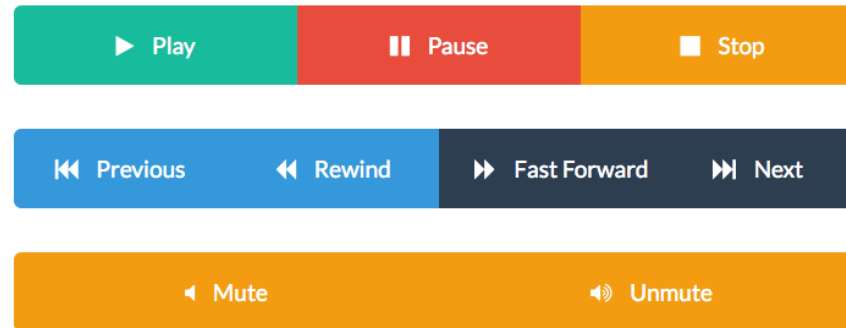
asgn2-desktop.png

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

CSCI 4140: Assignment 2

YouTube Remote

YouTube Remote Control



Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)



h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)



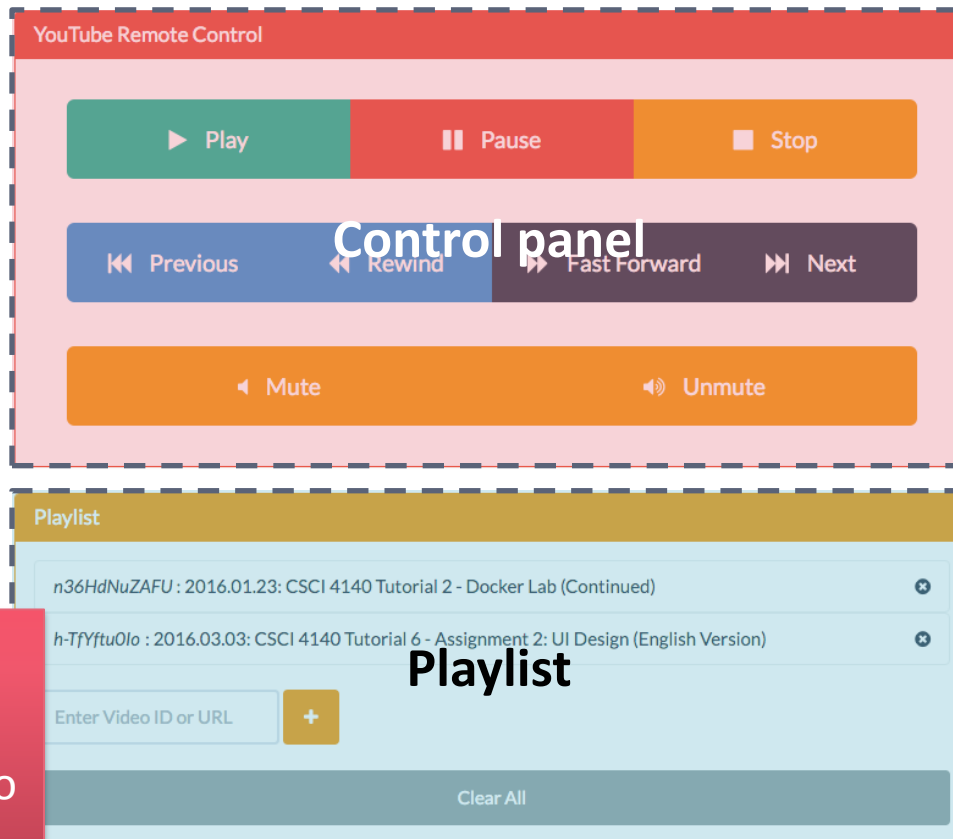
Enter Video ID or URL



Clear All

asgn2-tablet.png

CSCI 4140: Assignment 2
YouTube Remote



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

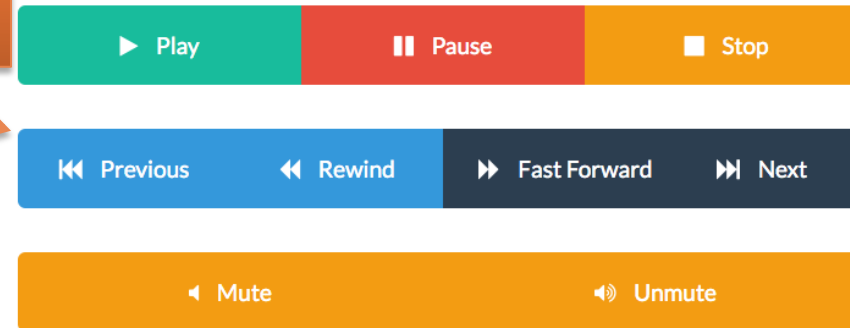
Note: When we reduce the width of the window, the video player is **removed**.

asgn2-tablet.png

The buttons are displayed with both icon and description.

CSCI 4140: Assignment 2 YouTube Remote

YouTube Remote Control



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

Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)

Enter Video ID or URL



Clear All

asgn2-tablet.png

When screen size is < 768px...

CSCI 4140: Assignment 2

YouTube Remote

YouTube Remote Control



Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)



h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)



Enter Video ID or URL



Clear All

asgn2-phone.png

CSCI 4140: Assignment 2

YouTube Remote

YouTube Remote Control



Control panel



Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)

h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English)

Playlist

Enter Video ID or URL



Clear All

When screen size is < 768px...

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

asgn2-phone.png

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

When screen size is < 768px...

CSCI 4140: Assignment 2

YouTube Remote

YouTube Remote Control



Playlist

n36HdNuZAFU : 2016.01.23: CSCI 4140 Tutorial 2 - Docker Lab (Continued)



h-TfYftu0lo : 2016.03.03: CSCI 4140 Tutorial 6 - Assignment 2: UI Design (English Version)



Enter Video ID or URL



Clear All

asgn2-phone.png

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



Client side: QR code display

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



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 a new video (with its video ID) to the playlist **(1)**
 - Remove an 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
 - **Method 1: Use DOM scripting**
 - Useful functions:
 - `document.getElementById()`
 - `document.querySelector()` (in HTML5)
 - `createElement()`
 - `appendChild()`
 - `removeChild()`
 - Read the lecture notes: “JavaScript (Part 1)”
 - **Method 2: Use React / Backbone.js**
 - Please refer to the lecture / tutorial next week

Client side: Playlist management

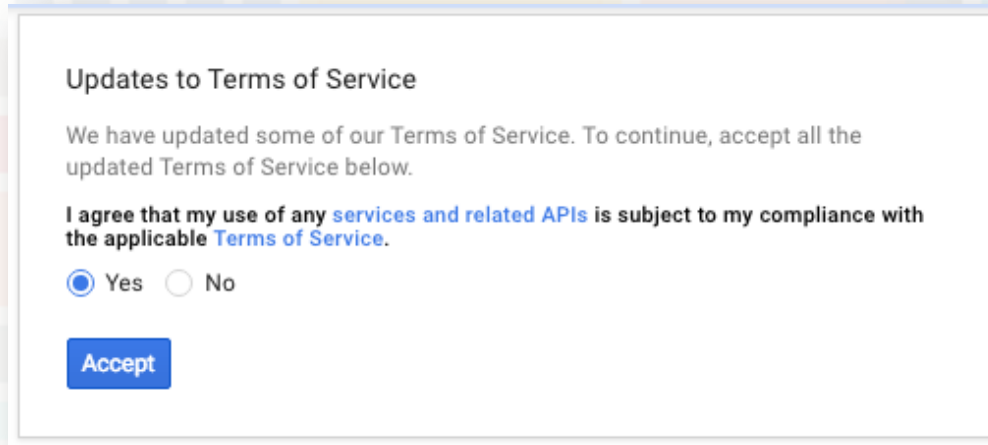
- To add a video, the user either input the video ID or the URL
- To retrieve the video ID from the URL, use this function:

```
function getVideoIDFromURL( url ) {  
    var regExp =  
    /^.*(youtu.be\/|v\/|e\/|u\/\w+\/|embed\/|v=)([^\#&\?]*).*;/;  
    var match = url.match( regExp );  
    // console.log( match[ 2 ] );  
    if ( match && match[ 2 ].length == 11 ){  
        return match[ 2 ];  
    } else {  
        return url;  
    }  
}
```

`client/get_video_id_from_url.js`

Client side: Playlist management

- You need to display the video title for each video in the playlist
- To retrieve the video title from YouTube, use **YouTube Data API**
 - Reference: <https://developers.google.com/youtube/v3/getting-started#intro>
 - **Step 1.** Login Google Developers Console (<https://console.developers.google.com/>)



Updates to Terms of Service

We have updated some of our Terms of Service. To continue, accept all the updated Terms of Service below.

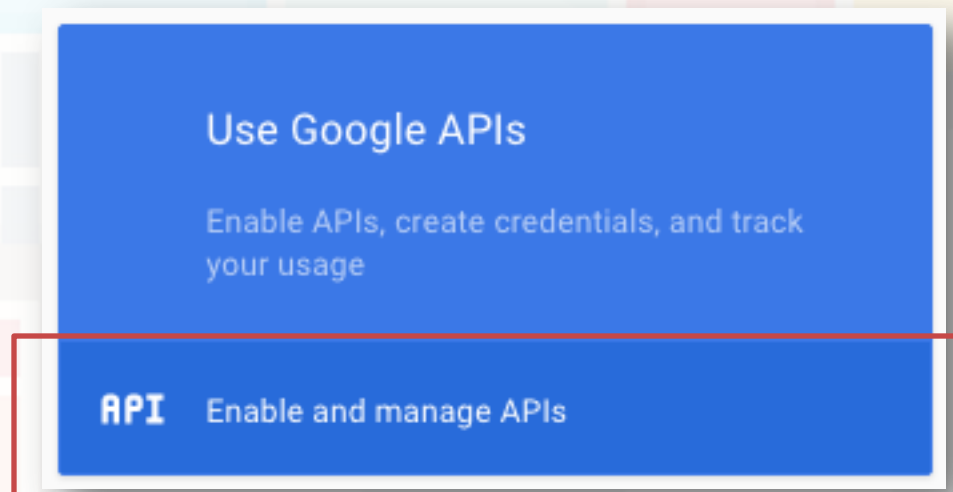
I agree that my use of any [services and related APIs](#) is subject to my compliance with the applicable [Terms of Service](#).

☒ Yes ☐ No

[Accept](#)

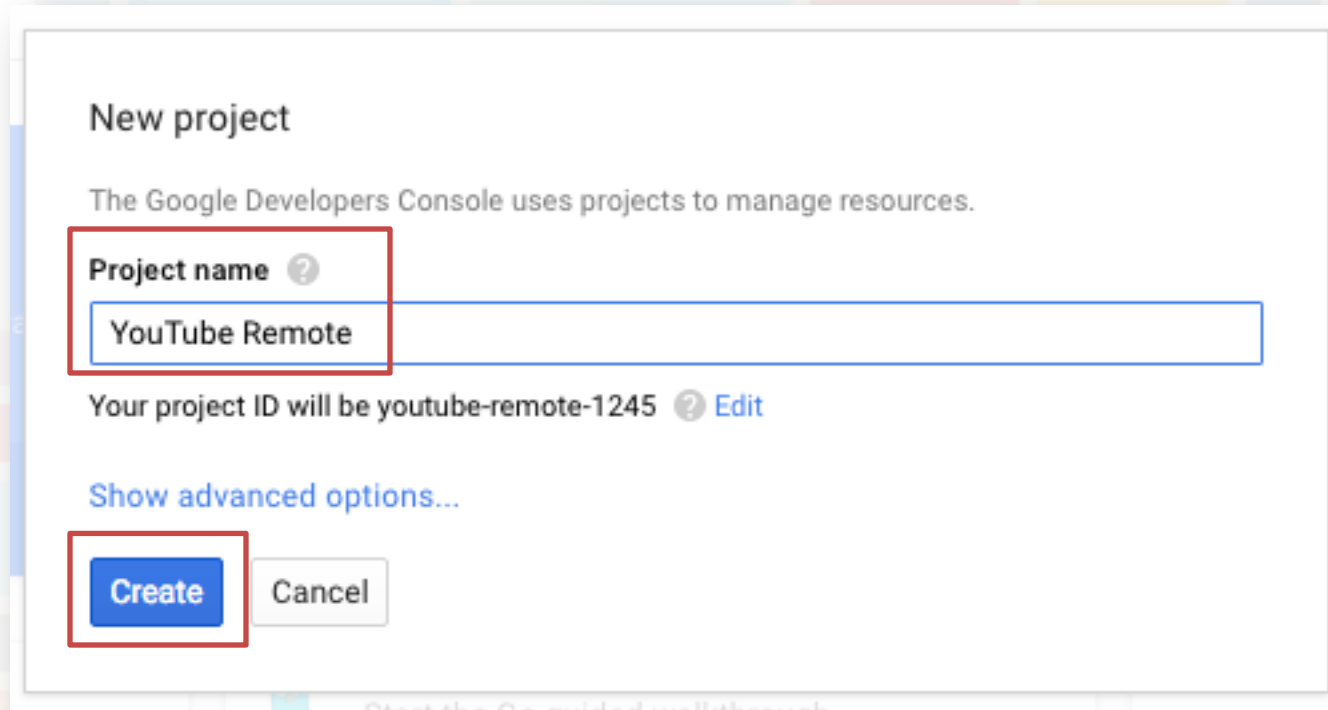
Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 2. **Enable Google API** in the Google Developers Console



Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 3. **Create a project** in the Google Developers Console



New project

The Google Developers Console uses projects to manage resources.

Project name ?

YouTube Remote

Your project ID will be youtube-remote-1245 ? [Edit](#)

[Show advanced options...](#)

Create Cancel

Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 4. Enable **YouTube Data API** in the project console

Popular APIs



Google Cloud APIs

[Compute Engine API](#)

[BigQuery API](#)

[Cloud Storage Service](#)

[Cloud Datastore API](#)

[Cloud Deployment Manager API](#)

[Cloud DNS API](#)

[More](#)



Social APIs

[Google+ API](#)

[Blogger API](#)

[Google+ Pages API](#)

[Google+ Domains API](#)



Enable

YouTube Data API v3

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists, and channels.

[Learn more](#)

[Try this API in APIs Explorer](#)

[Google Maps Roads API](#)

[More](#)



YouTube APIs

[YouTube Data API](#)

[YouTube Analytics API](#)

[YouTube Reporting API](#)

Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 5. Obtain authorization credentials for YouTube Data API

YouTube Data API v3

⚠ This API is enabled, but you can't use it in your project until you create credentials.
Click "Go to Credentials" to do this now (strongly recommended).

[Go to Credentials](#)

Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - **Step 5.** Obtain authorization credentials for YouTube Data API
 - Select “**Web server (e.g., node.js, Tomcat)**”
 - Select “**Public data**”
 - Click “**What credentials do I need?**”

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials

If you wish you can skip this step and create an [API key](#), [client ID](#), or [service account](#)

Which API are you using?

Determines what kind of credentials you need.

YouTube Data API v3

Where will you be calling the API from?

Determines which settings you'll need to configure.

Web server (e.g. node.js, Tomcat)

What data will you be accessing?

☒ Public data

Access publicly available data provided by the API

☐ User data

Access data belonging to a Google user, with their permission

What credentials do I need?

Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 5. Obtain authorization credentials for YouTube Data API
 - Click “**Create API key**”

2 Create an API key

This key should be kept secret on your server

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's `userIp` parameter, if specified. If the `userIp` parameter is missing, your machine's IP address will be used instead. [Learn more](#)

Name

Accept requests from these server IP addresses (Optional)

Examples: 192.168.0.1, 172.16.0.0/12, 2001:db8::1 or 2001:db8::/64

Create API key

Client side: Playlist management

- To retrieve the video title from YouTube, use **YouTube Data API**
 - Step 6. Done!

3 Get your credentials

Here is your API key

AIzaSyB1g6UkQWb5Lz3DPp4u5E-8YzP88DgKc

- Now, use your API key to send a GET request to the following URL:

`https://www.googleapis.com/youtube/v3/videos?part=id%2Csnippet&id=[Video ID]&key=[Your API Key]`

- The response is in **JSON format**
- **Self-study:** Read the example code “**youtube-data-api/index.js**” to see how to retrieve the video title

Client side: Playlist management

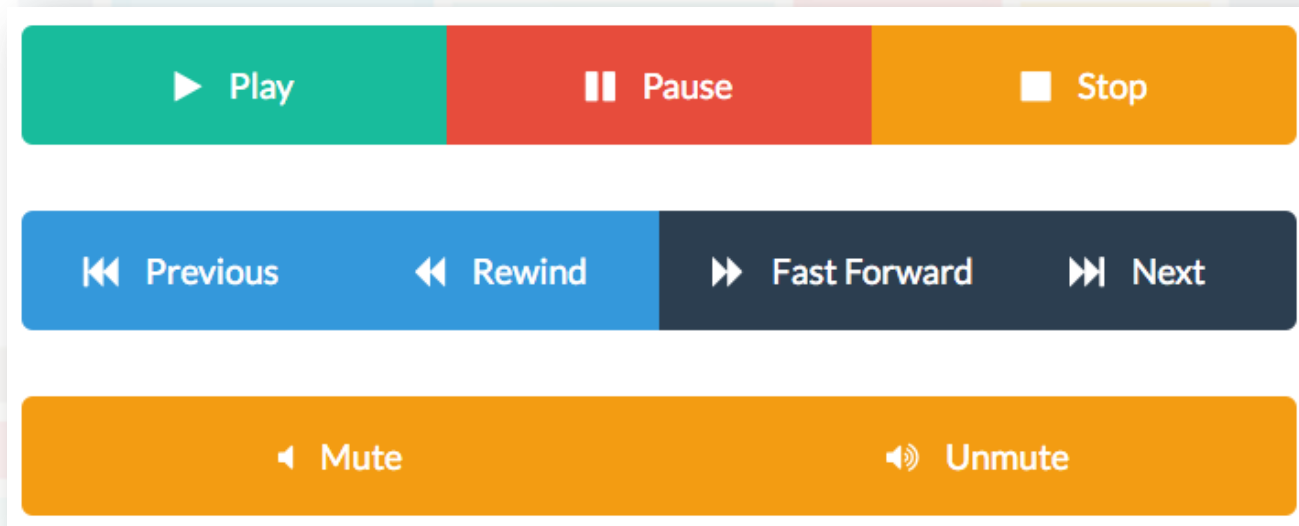
- You need to store the playlist on the server
- Since we have not covered server-side development yet, you may hard code the playlist on client-side for now

An illustration of a laptop, a tablet, and a smartphone. The laptop screen is filled with various colored rectangular blocks (blue, orange, red, yellow, grey) arranged in a grid-like pattern. The tablet and smartphone also display similar colored blocks. The background is white with a light green triangle on the left side.

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

Reminder

- You can start implementing the **UI**, **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

– End –