

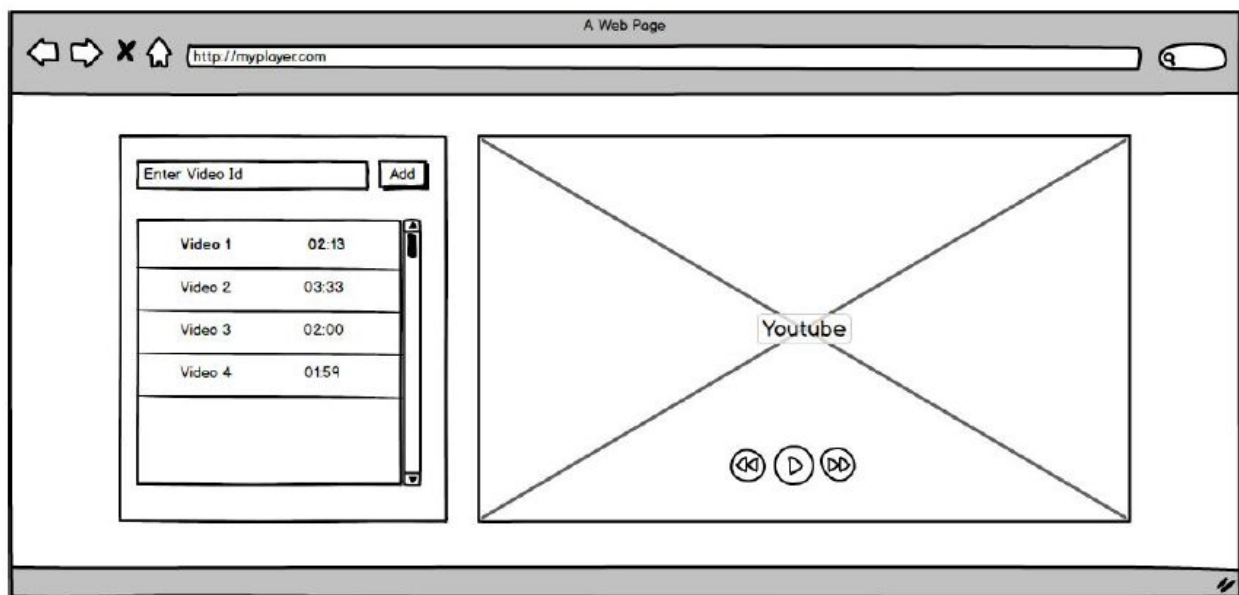
# Fullstack Home Assignment

Thank you for taking your time to complete this assignment.

The task is to create a web interface that allows the user to create a playlist of YouTube videos, and plays them in order and a server backend, which stores, and serves the playlist.

*Make sure to read the entire document before you begin designing and implementing your solution.*

Create the UI as specified in the wireframe:



# User Interface

The UI is made up of these components:

- An input field that accepts YouTube video URLs.
- A list of YouTube videos.
- A YouTube player, controlled by the list.

## Functionality

1. Start with an empty playlist, and nothing playing in the player.
  2. Any user can add a video by inserting text into the input field and pressing enter / clicking add.
  3. After adding the video, it is **added to the bottom** of the list. (showing its title and length is a bonus, you can use YouTube API - otherwise show the full URL).
  4. The player plays the videos in the order they were added. No way to change the order/skip/fast forward/rewind/etc...
  5. Once a video finishes playing, it is **removed** from the list, and the next video will begin playing.
- Mind that:
    - when the playlist is empty and the first song is added, it's immediately starts to play.
    - If a client joined when there's an active playlist, the player will play the first song in the list, from the beginning. The point is, that the video time shouldn't be synced across clients.

## Bonus

1. Add the ability to remove an item from the list.
2. Drag and drop to change the order of items on the list.

# Server Functionality

- Add backend functionality that stores/serves the playlist.
- Because the first item of the playlist is the same for all clients, effectively all clients see the same playlist.
- Once a user added a video, other clients should see the addition.
- Eventual consistency is more than enough.

## Scale

In your design - assume there are very few playlist editors, but many viewers. Your solution should be deployed to a single instance, but the design and implementation must demonstrate how this could be scaled out to support a very large number of concurrent requests. If there are any programmatic changes needed that will take a long time, you can just describe them in comments/separate text.

## Note that

This assignment shouldn't take more than a few hours. Feel free to make logical assumptions that makes sense, and make sure to document them so we'll understand the flow.

## Implementation

Please implement the UI with a high-level UI framework (React, Vue, Angular, AngularJS), written with **tests**. Backend technology of your choice. You can push your code to a public git repo, or send the directory as a zip (please make sure to remove `node_modules` before zipping). Add a short readme on how to **build and run** the project (code must be working).

**Feel free to use an open-source implementation for the YouTube player.**