

<Appendix>

First main page

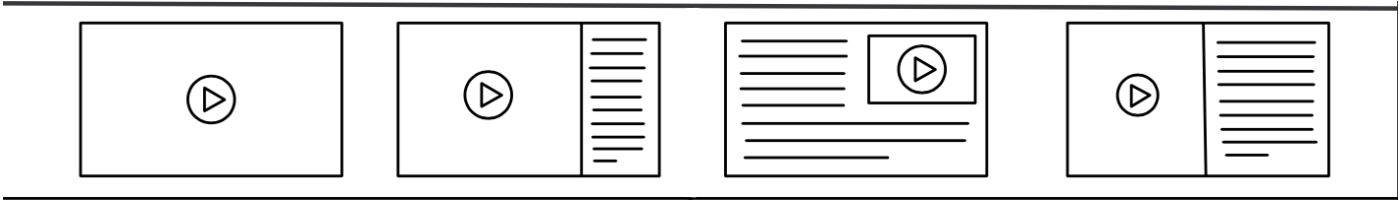
A wireframe of a web browser window. The browser has three tabs, a back arrow, a forward arrow, and an address bar. Below the browser controls is a toolbar with five buttons: 'OPEN FILE', 'SAVE', '<CODE>', 'RESIZE', and 'STYLE'. The main content area is split into two columns. The left column contains the text 'Enter your video link below' above a text input field, which is above a button labeled 'Go to your video'. The right column contains a large rectangular box with the text 'YOU NOTE' in a bold, sans-serif font, where 'YOU' is red and 'NOTE' is black.

Default Video + Note page

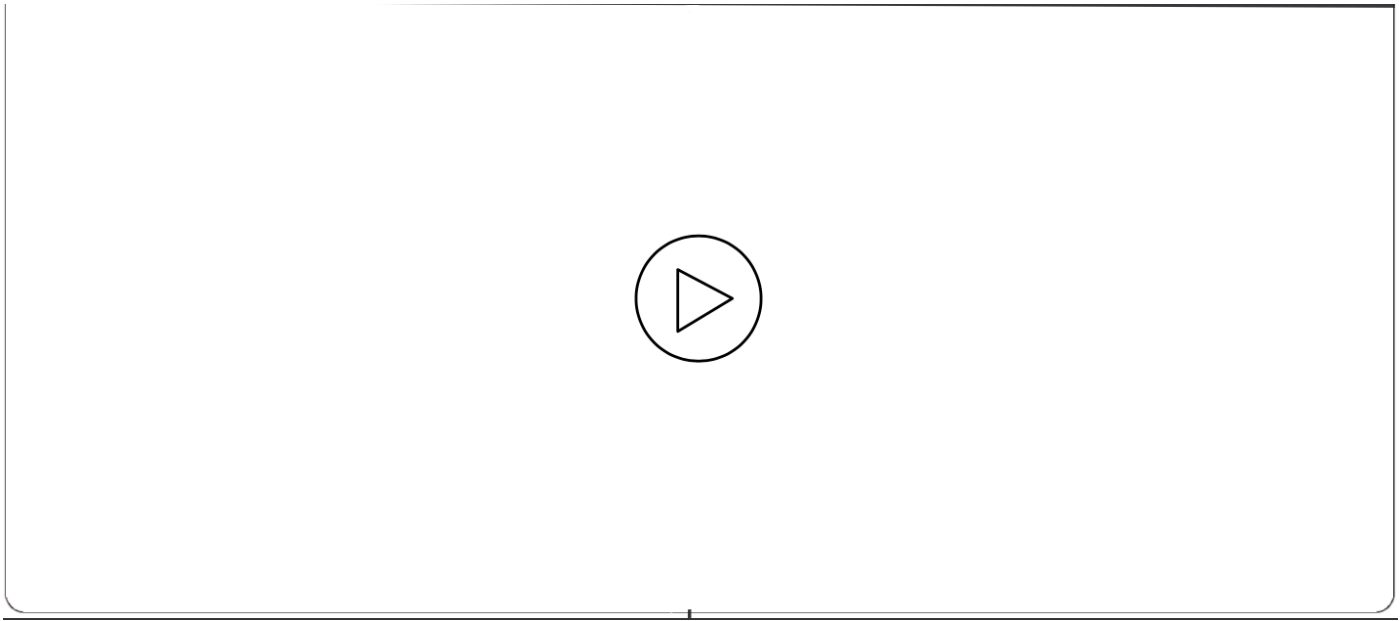
A wireframe of a web browser window, similar to the one above. It has the same browser controls and toolbar. The main content area is split into two columns. The left column is divided into three horizontal sections. The middle section is the largest and contains a large play button icon (a triangle inside a circle). The top and bottom sections are smaller and are currently empty. The right column is a single large empty rectangular area.

Type A – 1. Components for Resizing Video feature

Four different types of video resizing options



Video Resizing Option 1



Video Resizing Option 2

	<p>Network CSE310</p> <ul style="list-style-type: none">• Layering 7 layer is important• Performance How to measure the performance• EncodeFraming CRC, Parity bit• ErrorCorrection Check error
--	---

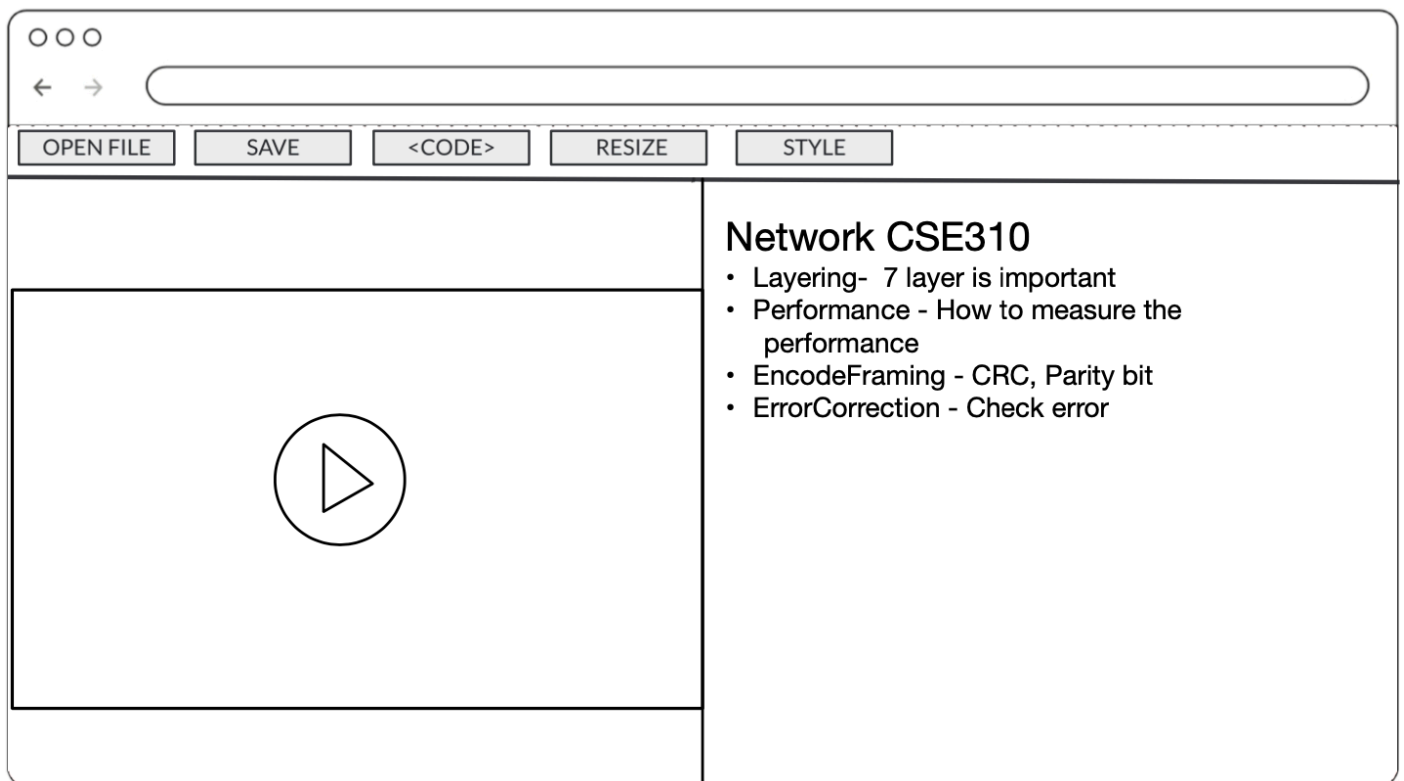
Network CSE310

- Layering
 - 7 layer is important because one vender cannot make everything. So it should be open for everyone.
- Performance
 - How to measure the the performance
- EncodeFraming
 - CRC, Parity bit
 - if something wrong, we should have ability to detect with CRC.
- ErrorCorrection
 - We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.



Type A – 2. Code Block Features

Page without code block



Black Code Block

```
1  
2  
3  
4  
5
```

Filled Code Block

```
1 def print_hi(name)  
2   puts "Hi, #{name}"  
3 end  
4 print_hi('Tom')  
5 #=> prints 'Hi, Tom' to STDOUT.
```

Note After Code Block

Network CSE310

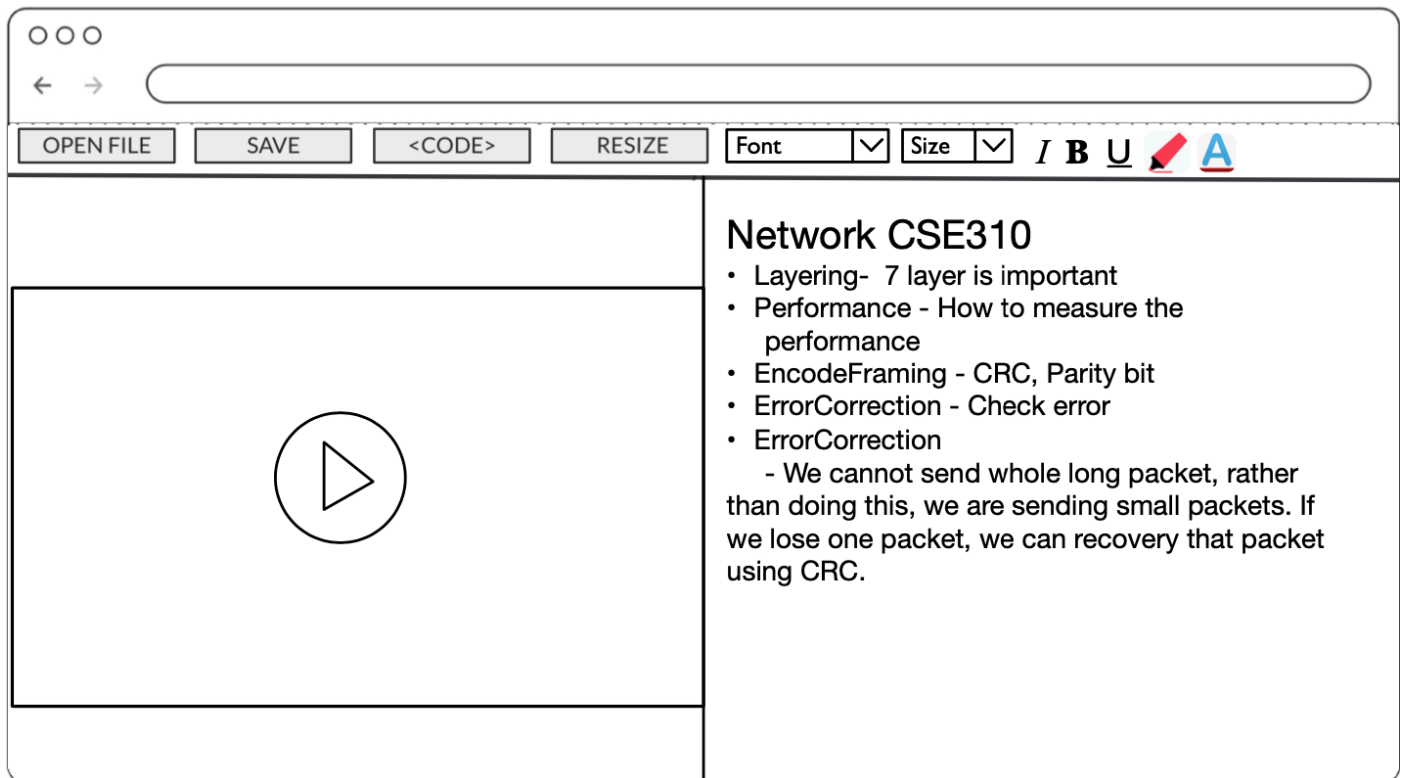
- Layering- 7 layer is important
- Performance - How to measure the performance
- EncodeFraming - CRC, Parity bit
- ErrorCorrection - Check error

```
1 def print_hi(name)  
2   puts "Hi, #{name}"  
3 end  
4 print_hi('Tom')  
5 #=> prints 'Hi, Tom' to STDOUT.
```

- ErrorCorrection
 - We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.

Type A – 3. Styling features

First page for font-styling



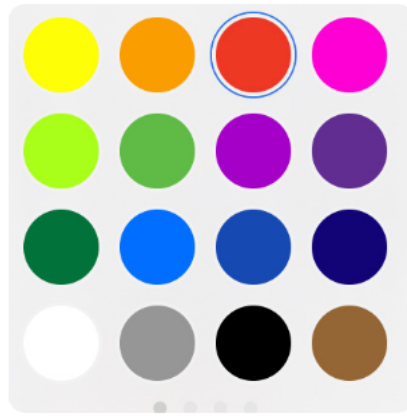
Changed font style text

- We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.

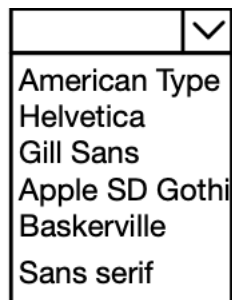
Changed font color text

- ErrorCorrection
 - We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.

Color options



Font-style choice

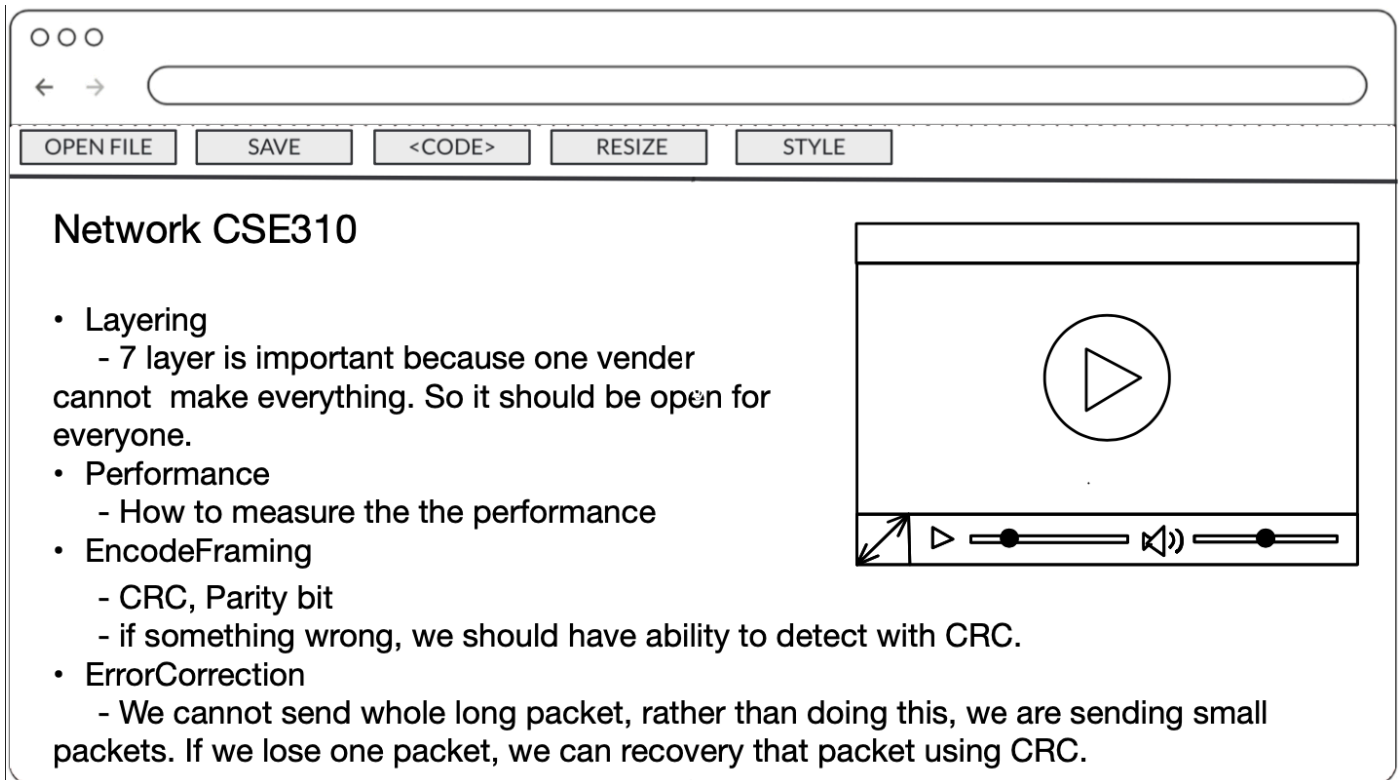


Dragged text with mouse

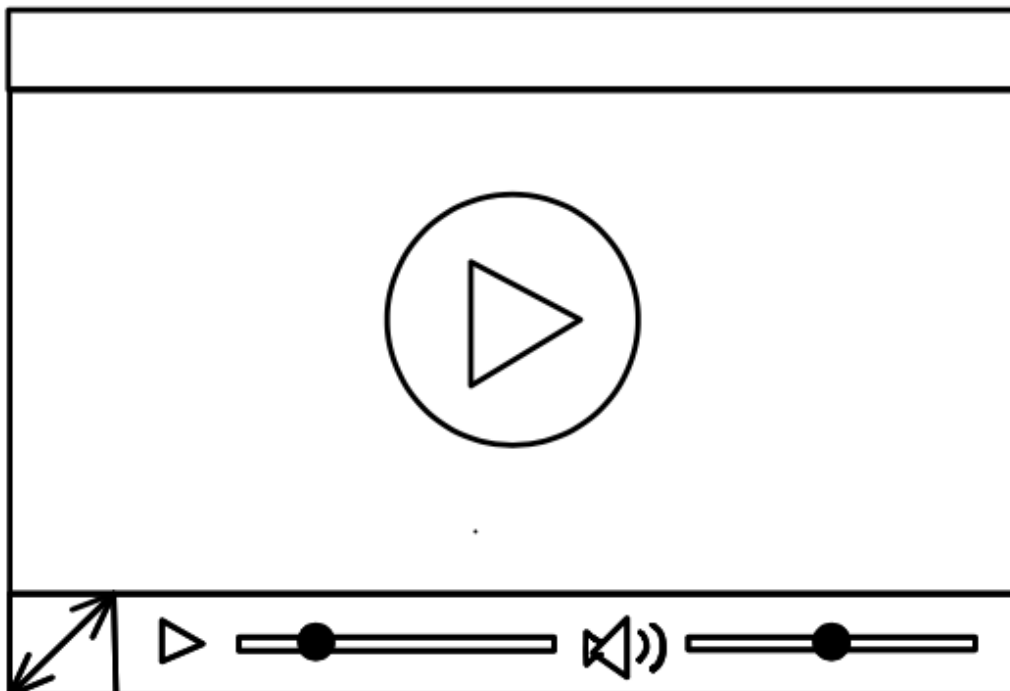
- **ErrorCorrection**
 - We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.

Type B – 1. Video Resizing Feature

Video Resizing Page

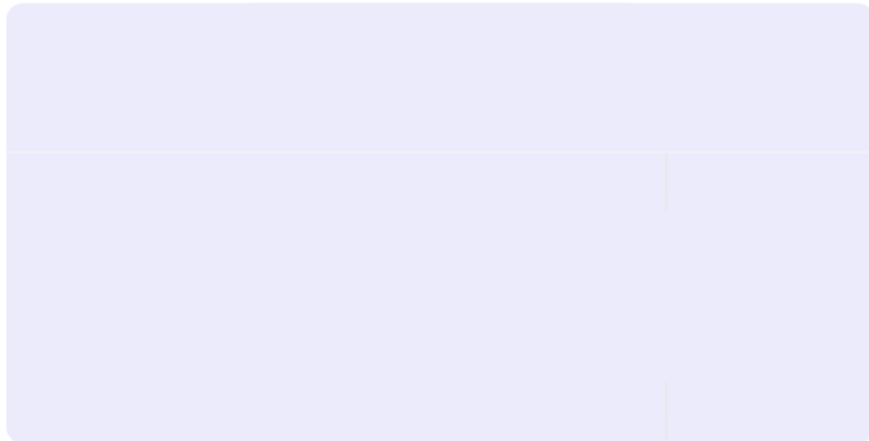


Separate Video Window in Video Resizing Mode

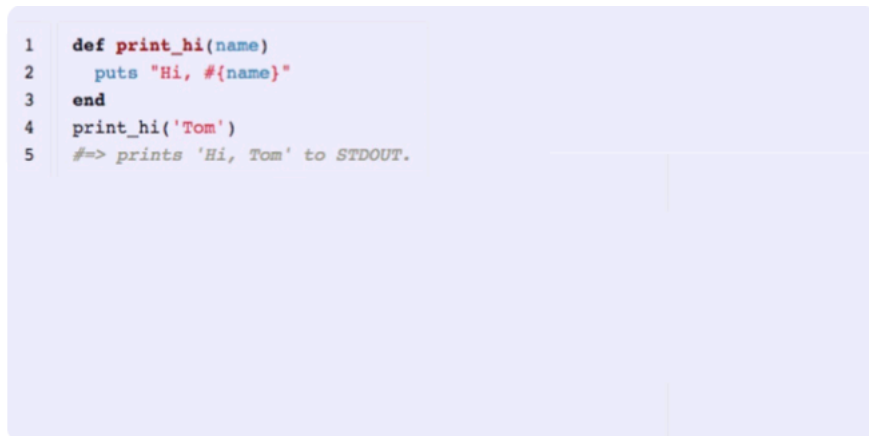


Type B – 2. Code Block Features

Black Code Block



Filled Code Block



Note After Code Block

Network CSE310

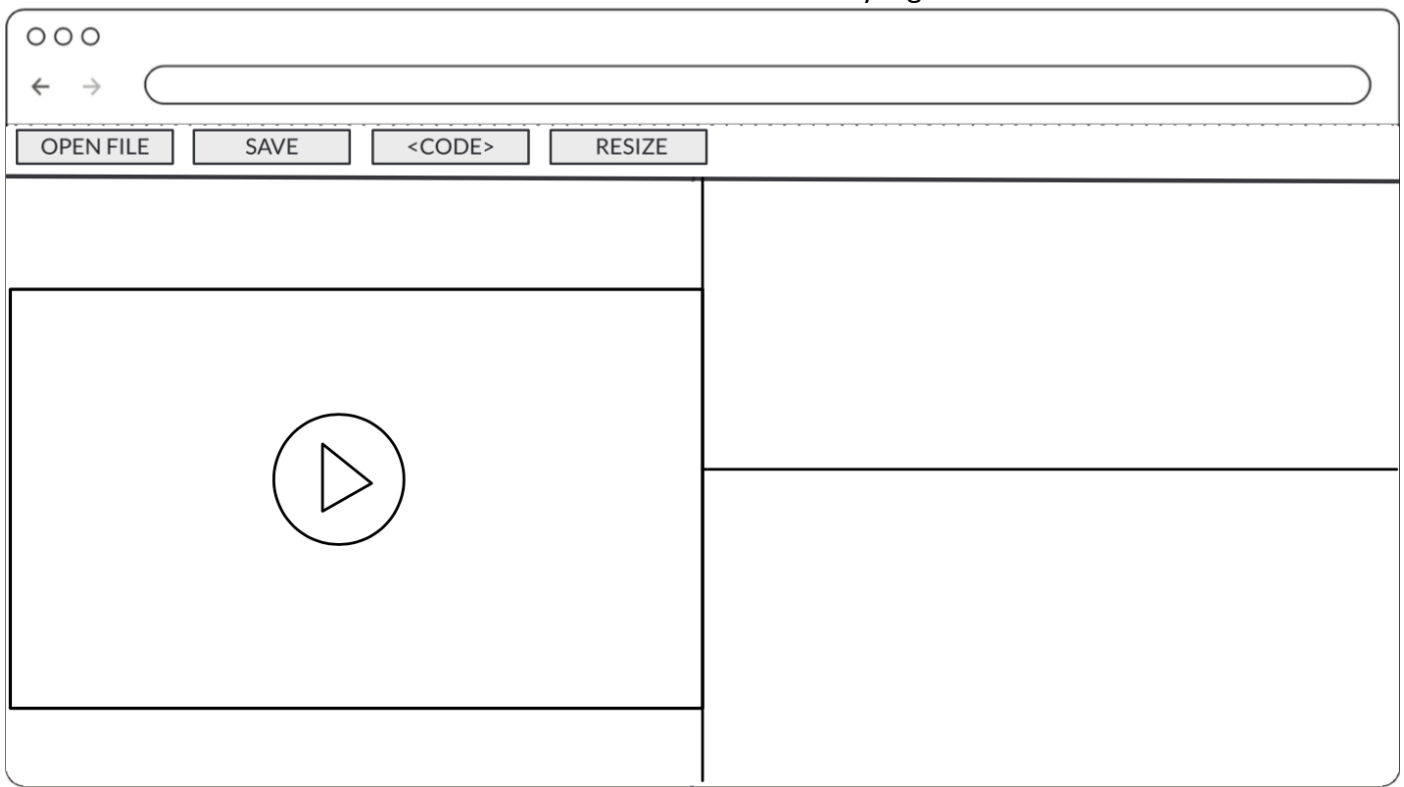
- Layering- 7 layer is important
- Performance - How to measure the performance
- EncodeFraming - CRC, Parity bit
- ErrorCorrection - Check error

```
1 def print_hi(name)
2   puts "Hi, #{name}"
3 end
4 print_hi('Tom')
5 #=> prints 'Hi, Tom' to STDOUT.
```

- ErrorCorrection
 - We cannot send whole long packet, rather than doing this, we are sending small packets. If we lose one packet, we can recovery that packet using CRC.

Type B – 3. Font Styling with Markdown syntax

blank boxes for Markdown styling



Markdown syntax

```
1- ---
2 title: "Simple Language Demos"
3 output: html_document
4 ---
5
6 You can write code in languages other than R with R Markdown, e.g.
7
8 ## Bash
9
10 ```{bash}
11 ls *.Rmd
12 ```
```

Rendered Markdown page

Simple Language Demos

You can write code in languages other than R with R Markdown, e.g.

Bash

```
ls *.Rmd
```

```
## 1-example.Rmd
## 2-chunks.Rmd
## 3-inline.Rmd
## 4-languages.Rmd
```