Test Project

Web Technologies

Speed Test- Front End

Submitted by:

Trần Đức Doanh
Lương Văn Hiếu
Kiều Thành Chung
Kim Thị Thương
Trần Việt Khánh
Nguyễn Ngọc Anh
Khúc Thị Ngọc Hà

Contents

Introduction

In this module, you are given 2.0 hours to complete mini tasks. There are 3 parts of mini tasks, they are design, layout and front-end development. 1.5 point for each easy task, 2 points for each normal task and 3 points for difficult task.

You do not need to do all of the mini tasks. You will receive instruction on which tasks that is required during competition by random drawing on day C-2.

Part 1: Website Design

1.1. Blur Picture Image (Easy)

Given a picture image, please use graphical software to make this on greyscale, put a blur effect, and save as a1.jpg.

1.2.Transparent Background (Easy)

Given an image on format jpg. Please use graphical software to turn the background into transparent, and save as a2.png.

1.3. Pattern (Easy)

Make a background image with any line pattern that repeat every 100x100 pixels, and save with 1920x1080 px as result.jpg.

1.4. Loading GIF (Normal)

Create a GIF animation with a word “Loading” that appears on the left edge of image and moves outside of the right edge of image. The animation should be within 3 seconds to complete.

1.5. UI-kit (Difficult)

Create a UI-kit of the following elements: text box, button, drop-down list, checkbox, radio button, label with checkbox, label with radio button, drop-down calendar input, range, progress bar.

1.6. Messenger (Difficult)

Draw 2 screen mockups for a messenger app with size 360x640.

1 screen - a list of dialogs (avatar, full name, message preview).

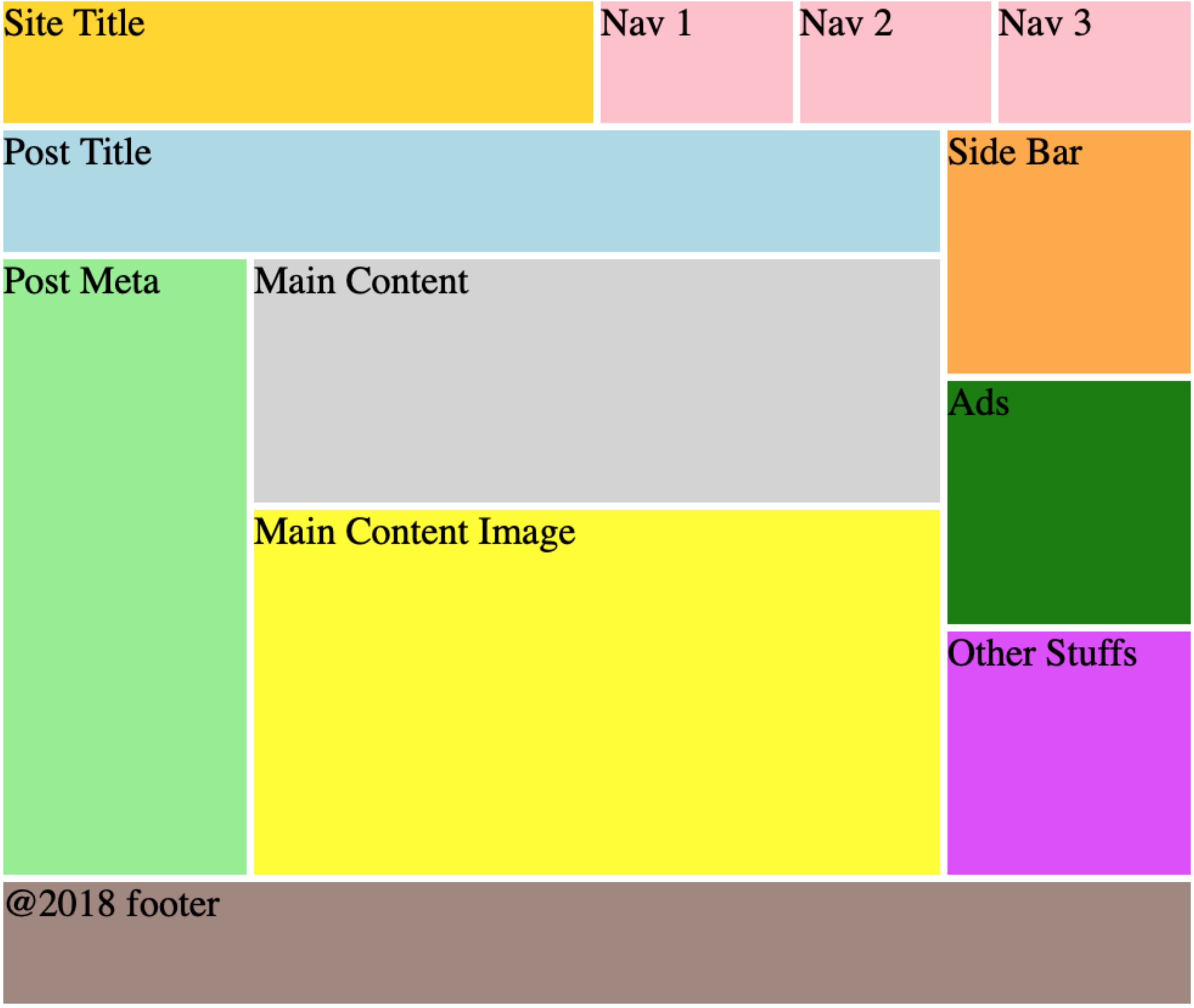
2 screen - chat with the user (avatar, full name, menu (three points), messages, text field, send button)

All elements should exist.

Part 2: Layout

2.1. CSS Grid (Normal)

Please implement the following layout in CSS grid.



Site Title

Nav 1

Nav 2

Nav 3

Post Title

Side Bar

Post Meta

Main Content

Ads

Main Content Image

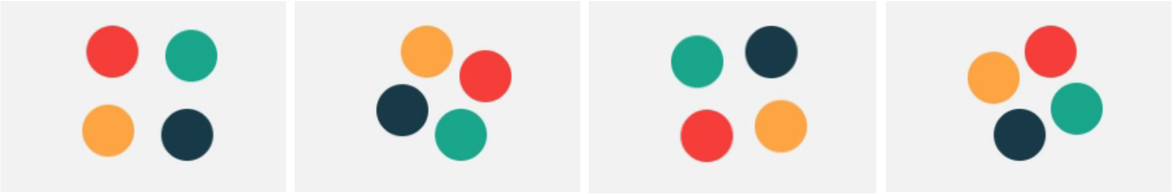
Other Stuffs

@2018 footer

2.2. Loading Screen (Normal)

Create an animated loader using only HTML and CSS same as video.mp4. The loader animation should be looped. 4 different colors (#19A68C, #F63D3A, #FDA543, #193B48) should be used.

Screen shots:



2.3. Toggle (Normal)

To create an IOS Toggle switch using HTML and CSS only. View video for animation details.

2.4. Triangle (Easy)

Make a blue equilater triangle on the center of the view, and animate this to transform a yellow square. the animation needs to be fluid and loop infinite;

2.5. Grayscale image (Easy)

- Turn the image on grayscale, and when hover the mouse, the image need to be color with a transition;

2.6. Jumper Ball (Easy)

- Make a loop animation of a jumper ball respecting all states from a animation (only css);

2.7. HTML/CSS – Contact Form (Normal)

An HTML and CSS implementation of a not yet finished form are given.

Modify the CSS and HTML so that the form looks as depicted in the screenshot in the media files.
It's responsive behaviour should follow the screencast in the media files.

The use of JavaScript is NOT allowed.

2.8. Border Radius & Box Shadow (Easy)

Create a page that contains a 400px rectangle in the middle.

Inside the rectangle there is a settings panel with inputs, where you can setup the rectangle's Border radius, and box shadow properties. The rectangle’s css updates on input changes.

2.9. Race track (Normal)

By using HTML and CSS, create a race track animation in the form of “8” (vertically or horizontally), along which the car will endlessly animated.

2.10. Video player interface (Layout)

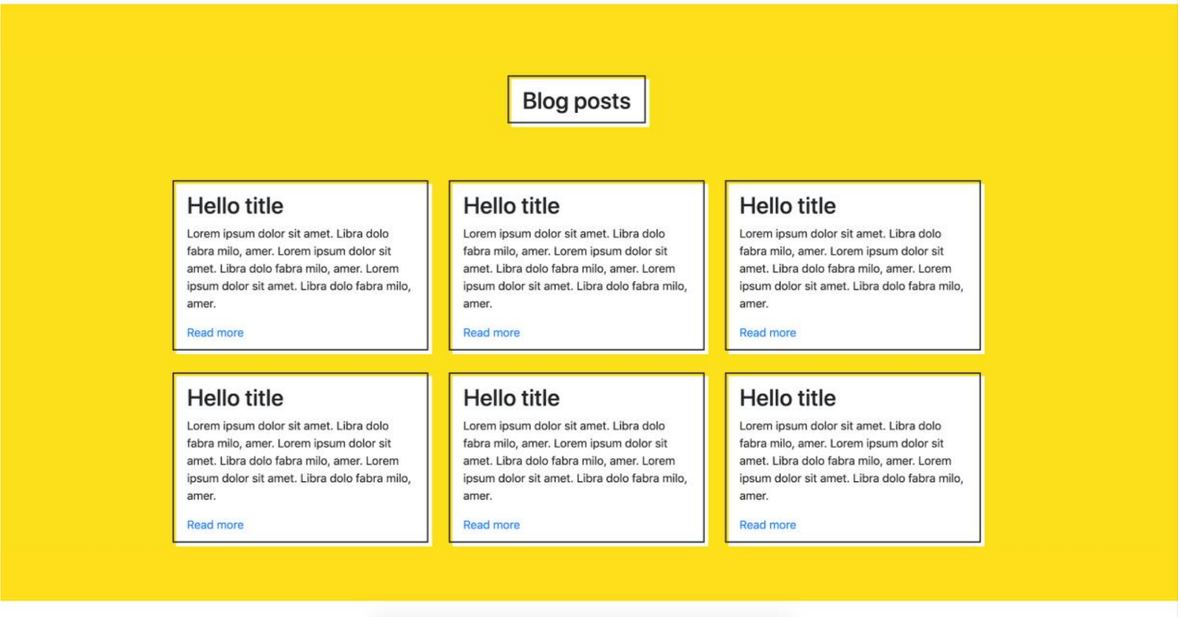
Create video player interface: search form, browse screen, action buttons (play, mute, and volume), progress slider, playlist (one item from playlist must be active), information about the video (title, description, number of views), comments (not less than 5) (and 1 attached comment).

2.11. Gradient animation (Easy)

By using HTML and CSS, create a gradient shift animation on the page. The gradient may move into any direction.

2.12 Modern layout (Easy)

Your task is to create the following layout, the posts’ width is ⅓ of the container’s on desktop screen size (>=1440px) and full width below this.



Blog posts

Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

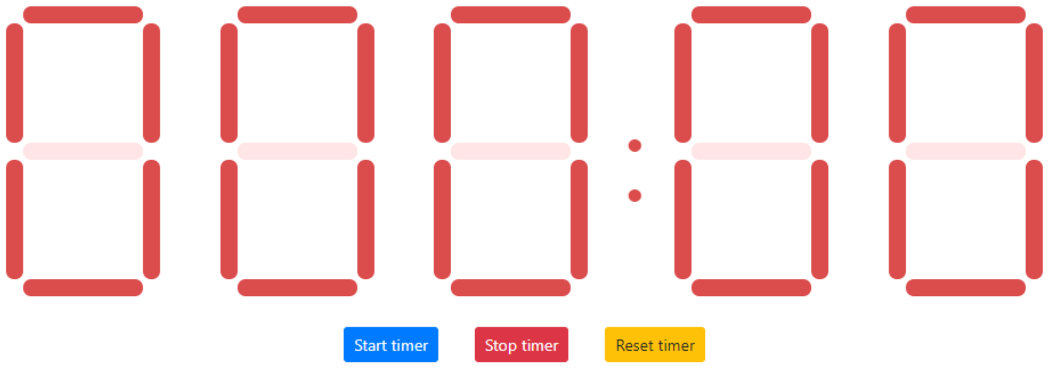
Hello title

Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer. Lorem ipsum dolor sit amet. Libra dolo fabra milo, amer.

Read more

2.13. Seven segment digital Timer (Difficult)

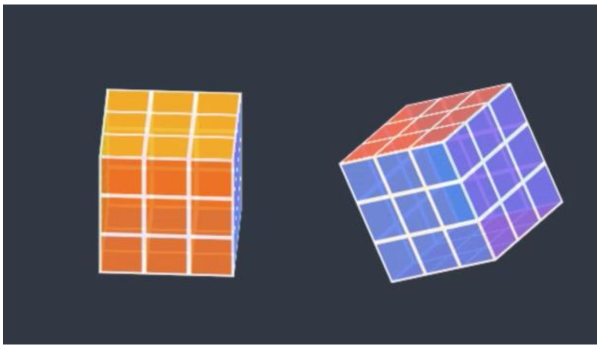
Create a 7 segment digital timer, time format is: 3 digit of seconds passed, and centiseconds following it. It’s max value is: “999:59”. “Start timer“ button starts, “Stop timer” button stops, “Reset timer” button stops and resets to 0 the timer.



Part 3: Front-End Development

3.1. Cube Rotation (Difficult)

We want to create the following cube rotation effect.



1. You should create two cubes using the index.html and the cubes shuld be rotated same as video.mp4.

2. Left cube should be rotated left to right.

3. Right cube should be rotated up to down.

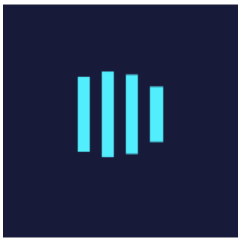
4. color codes

    - front - #fa5252
    - back - #f76707
    - right - #12b886
    - left - #4c6ef5
    - top - #fab005
    - bottom - #7950f2

5. You can only use style.css for the Cube Animation, JavaScript is not allowed.

3.2. Loading Animation (Normal)

We want to create the following CSS loading animation.



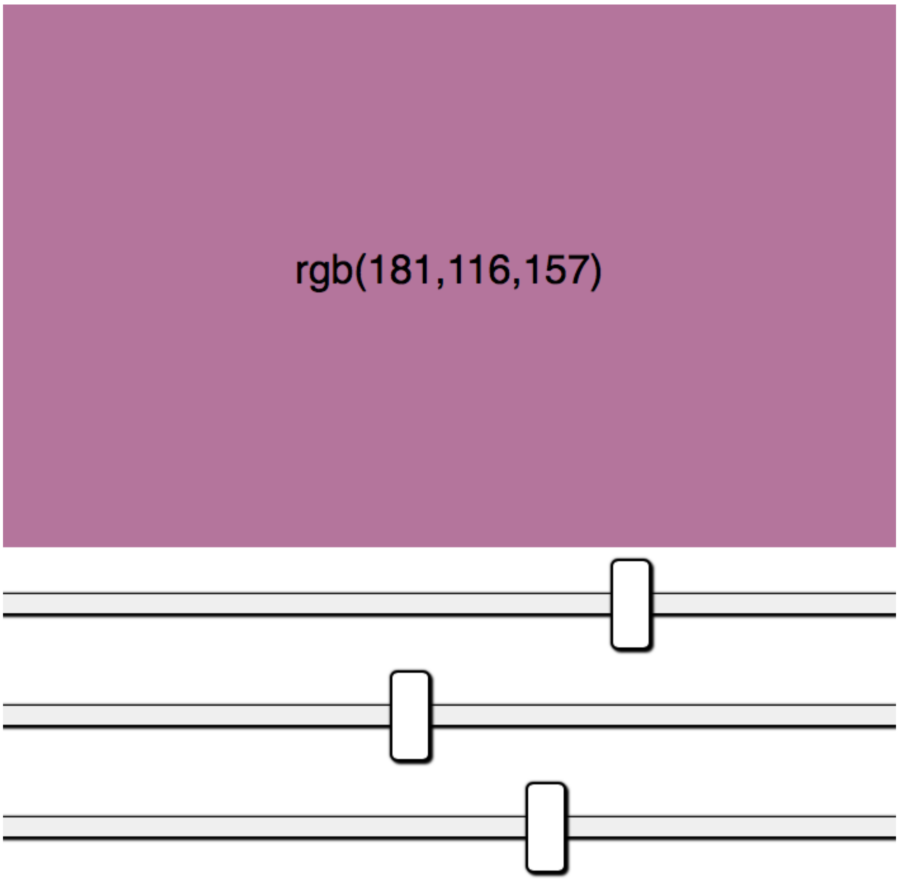
3.3. Drawing Canvas (Normal)

Please try to use canvas to build a drawing pad with the following features:

- Basic drawing with mouse down and move
- 3 color switching

3.4. RGB Slider (Normal)

Please implement an RGB slider as following. There are 3 sliders for adjusting values of red, green and blue.



3.5. JS – Calculation (Easy)

Implement the function that calculate the sum of the squares of the odd numbers in a given array.

3.6. JS - ASCII-Art (Difficult)

You should implement the two methods for transforming the ASCII-input (an image made of standard characters) as described in file “ascii-art-editor.js” (rotate and mirror).

You can check the functionality in index.html – it will execute tests (call your functions with different input and test the output) and show its results.

3.7. HTML/CSS, JS – Clock (Difficult)

You have to implement a clock showing the current local system time.

The background should look as depicted in the media file “full-clock-fancy.png” containing clock border and ticks for hour and minute. Also the clock hands should look as depicted in the image. Seconds clock hand is updated every second, minutes clock hand is updated every minute, hour clock hand is updated every minute.

The image “full-clock-fancy.png“ can be used as you wish.

3.8. List of expenses: (Easy)

3.8.1. Create a form with two fields called: **Spending name** and **Cost** and an **add** button.

3.8.2. Using javaScript, perform the following functions:

1. Get the values of the fields.
2. Validate the entry of the data, if the fields are empty, print the message "There was an error", otherwise print "Correct".
3. Remove the expanded message and the data entered in the form after 3 seconds.
4. The data after being entered, must be shown in an ordered list using the HTML tags.

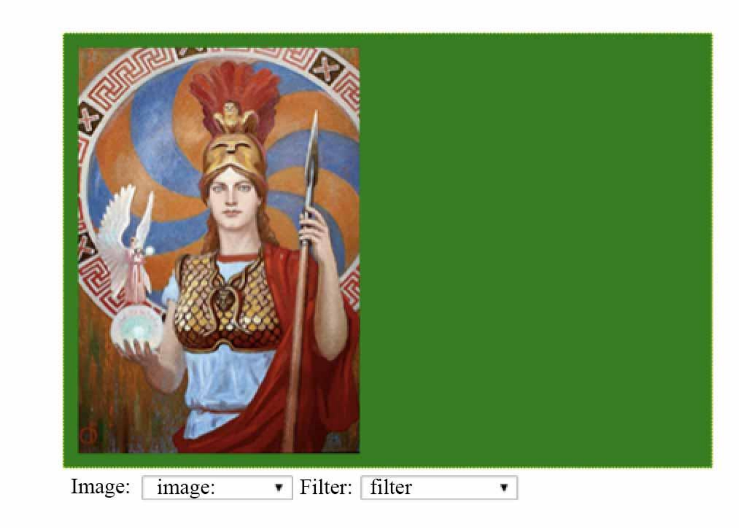
* No resources are provided.

3.9. Simple Image Filter (Difficult)

For this speed Project you must create a function that allows you to add a filter in a certain image using canvas. This speed Project is divided into two parts.

Part 1

- Develop the following design:

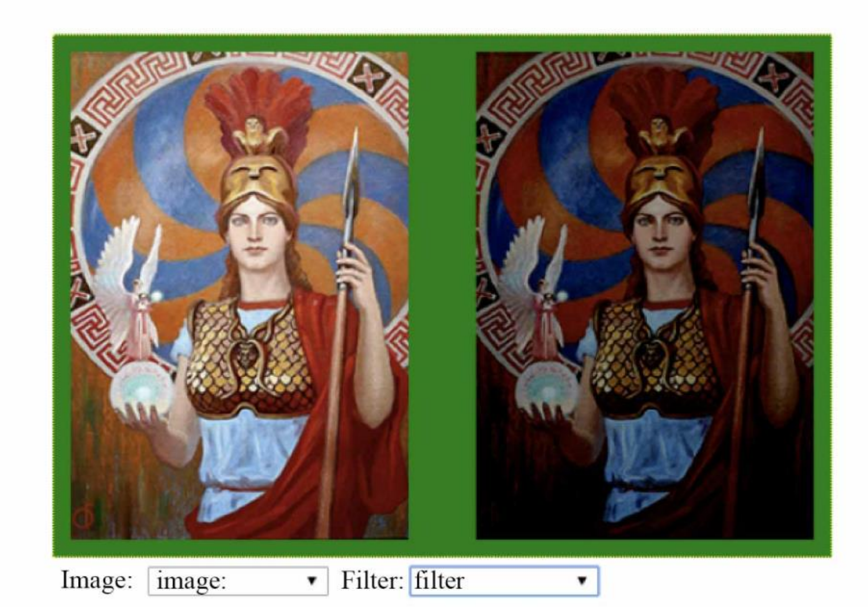


- The size of the canvas should be 460 pixels wide and 320 pixels high.

- Create a drop-down list, with the image options: Athena.jpg, theKiss.jpg, young-pearl.jpg and mona-lisa.jpg. There must be a first option called image.
- Load the images on the canvas depending on the option chosen in the drop-down list.
- The image of athena.jpg is loaded by default.

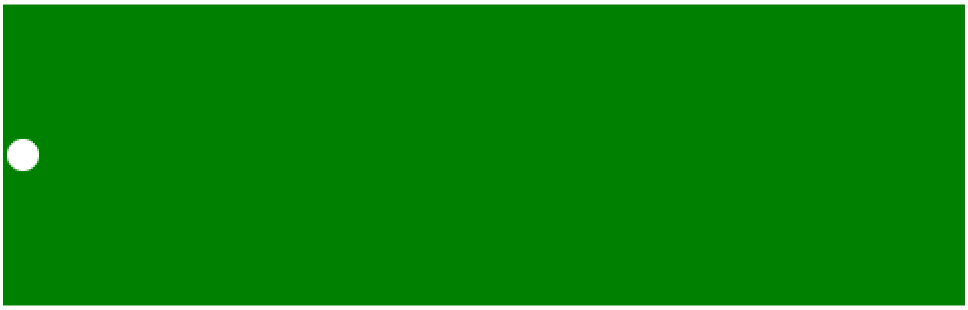
Part 2

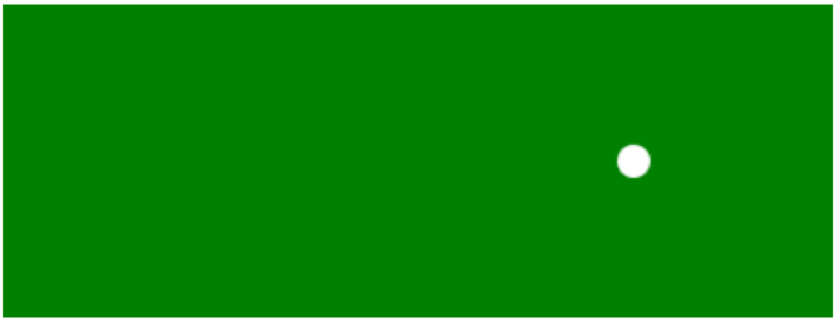
- Create a drop-down list, called filter with the options: Darken and Lighten. There must be a first option called filter.
- When choosing a specific image in the dropdown list and applying a filter, the result should appear on the right side of the canvas according to the chosen option.



3.10. Canvas Animation (Easy)

Draw a circular figure in a container of 320px height and 400px width using canvas. When the browser window reloads, the circular object must move from left to right continuously.





3.11. JS Arithmetic (Difficult)

Please create an arithmetic function, that when you enter a string of expression, it returns the calculated result. This function should include the following operators, listed by precedence:

a. Parentheses: ()

b. Minus sign: -

c. Exponentiation: ^

d. Multiplication, division, modulus: * / %

e. Addition, subtraction: + -

Example 1:

The function input: 1+2*3

The function output: 7

Example 2:

The function input: 3^2%5

The function output: 4

Example 3:

The function input: (-1-2)^3/9

The function output: -3

3.12. JS Draggable (Normal)

Please implement Draggable function by using provided script.js, to make the parameter element draggable. Please see the provided video named video.mp4 to find draggable effect.

3.13. Keyboard (Difficult)

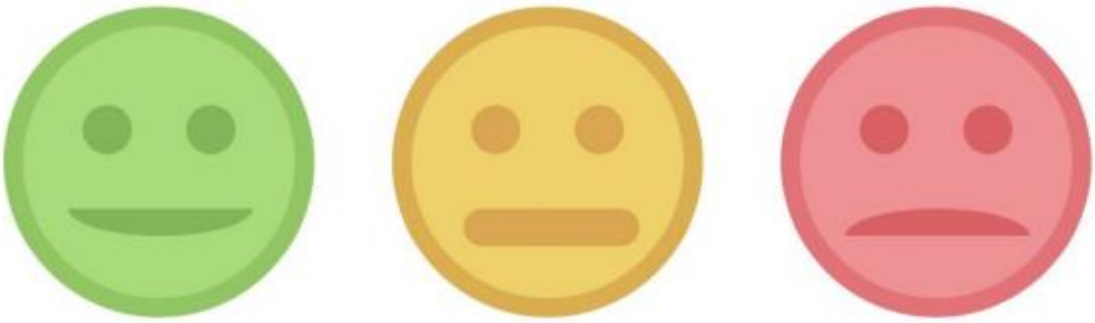
Create a text area where the text will be typed and under it a virtual keyboard on which keystrokes will be highlighted, including holding shift and stuff. Also with the keys on the virtual keyboard, you can type text by clicking on them. Shift is turned on and off by clicking on it, while the remaining keys enter their character in the text area.

3.14. Parallax (Normal)

Adjust the background parallax depending on the position of the mouse cursor (the closer the objects are to the user, the faster they move).

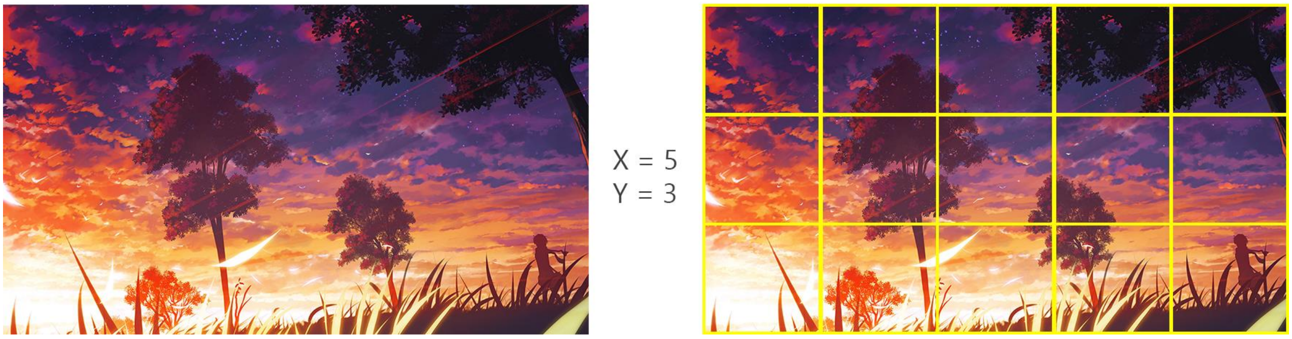
3.15. Smiley (Normal)

Create a big smiley face on the center of the page. It is 400px wide, and rectangular. Below the smiley, there are 3 buttons with texts: “:)”, “:|”, and” :(“. By clicking the buttons the smiley face changes to that state with a 0.5s animation. By default the smiley is in “:|” state. The states have these designs:



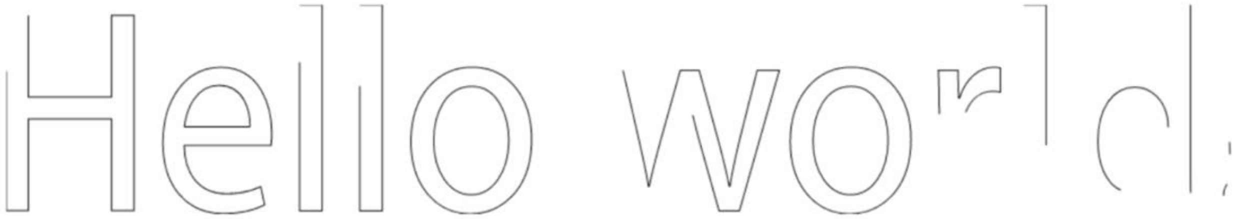
3.16. Transition (Normal)

You are given a picture that you need to cut into N cards and each card should disappear animated. X is 5 and Y is 3.



3.17. Animation (Normal)

Create the same animation that is presented in the video. A Hello world wording appears with line animation.

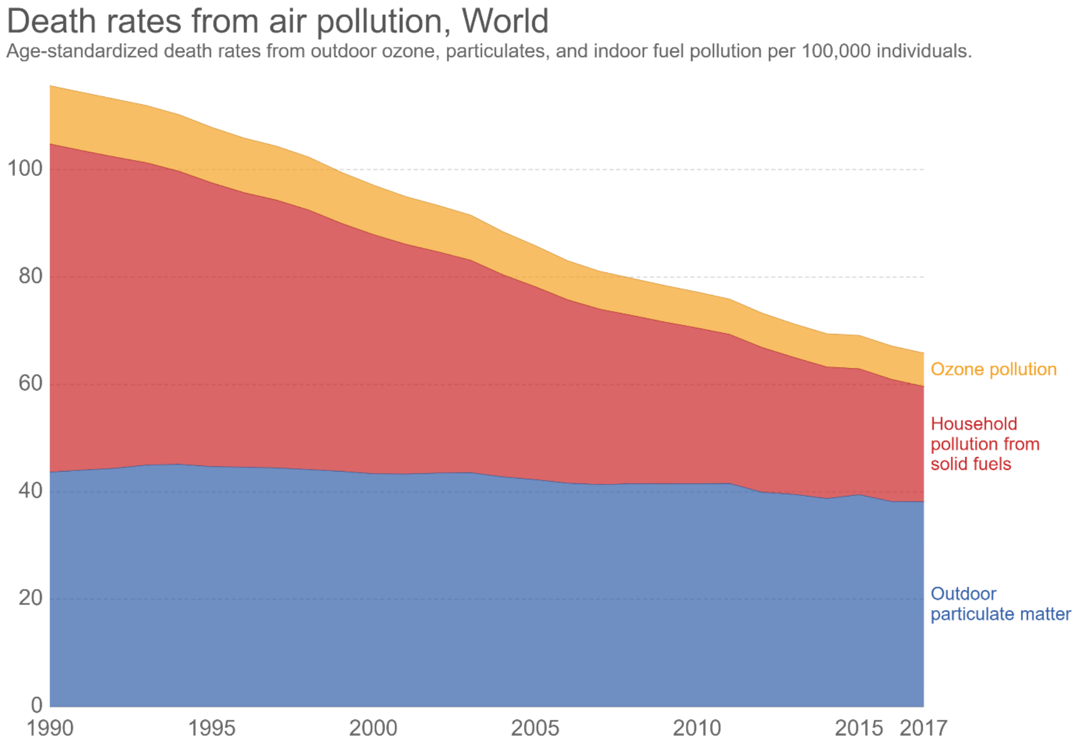


3.18. Chart from data (Difficult)

You should create a Chart Drawing with provided data(death-rate-by-source-from-air-pollution.csv). There are Death rates from air pollution of all countries from 1990 to 2017 in the data. Chart shoud calculate all data of all countries.

In the page, there shoud be a Chart which shows rates of 3 data (Ozone pollution, Householdpollution from solidfuels, Outdoor particulatematter) from 1990 to 2017. See the examples below, but you can create your own style.

For this test, you can only use HTML, CSS and JavaScript (including jQuery or JS Framework).



Death rates from air pollution, World

Age-standardized death rates from outdoor ozone, particulates, and indoor fuel pollution per 100,000 individuals.

3.19. Array Integers (Easy)

Given an array that contains integers, rearrange on crescent order only in the positives integers, keeping the negatives on the same position (consoles the new array rearranged);

3.20. Following cursor (Easy)

You have a div on the screen. make the div follows and be centered of accord a cursor position in the window;

3.21. Circles (Normal)

Create a page, that contains 5 circles. Each circle contains one of the following texts: Home, About, Contact, Map, Privacy, the texts are centered both horizontally, and vertically. Position of the circles are random on the page, but no overlap allowed. The whole area of the circle must be visible. The page has a 20px padding.. Size of the circles is random between 300-400px on each load too.

The color of the circles are: #1F2041, #4B3F72, #417B5A, #D0CEBA, #E9D2C0.

3.22. Linear gradient (Normal)

On the center of the screen there is a 300px sized square. By default it has a red to blue linear gradient background. On the left and right of the square there are 1-1 input fields where you can change the starting, and ending color of the linear gradient, by writing in the HEX value of the color. Over the inputs there are 10 randomized color buttons (10px*10px) on each side. If you click one of the randomized colors, it's color is inserted in the input. Whenever any input is changed, and is a valid hex, that side's linear gradient color changes on the square.

Instructions to the Competitor

Put all your done tasks on: http://localhost/XX_Module_A

XX is the competitor's workstation number.

Marking Scheme

SECTION	CRITERION	JUDGEMENT MARKS	MEASUREMENT MARKS	TOTAL
Part 1	Design	0	4	4
Part 2	Layout	0	4	4
Part 3	Front-end	0	4	4