worldskills

Test Project

Module A Mini Speed Test Projects

Web Technologies

Thomas Seng Hin Mak
Leyla Santos BR
Valentin Adamescu UK
Seung-Iyul Ryu KR
Zhang Ling CN
Ryoju Ohata JP
Ebben Kapali NA
Manuel J. Schaffner CH
Kok Leong Tang SG

WSC2024_TP17_en

Contents

Introduction

In this mini speed Test Project, you are given multiple mini tasks. We have collected all the tasks from multiple clients. And for priority reason, we have selected some tasks for this round of development. Please only implement the required tasks.

Please follow the instruction to know which one to implement.

For each mini Test Project, there are three levels.

- Level 1: 0.5 points worth that is expected takes less than 15 minutes of work.
- Level 2: 1.0 points worth that is expected takes less than 30 minutes of work.
- Level 3: 1.5 points worth that is expected takes around 30 minutes of work.

Please put your files in /XX_module_a/ and with the task ID as folder name, e.g. /XX_module_a/A1/

The XX is the workstation number.

Description of project and tasks

Please create an index page to link to each mini speed Test Project.

The index page should contain thumbnail and title of each mini speed test project.

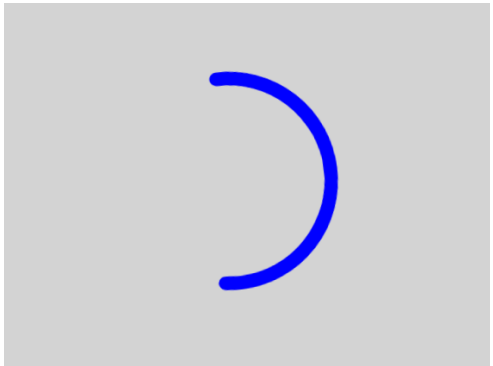
Please write at least one git commit for each mini speed Test Project. The git commit messages should be readable and easy to understand.

Tasks for Design Implementation

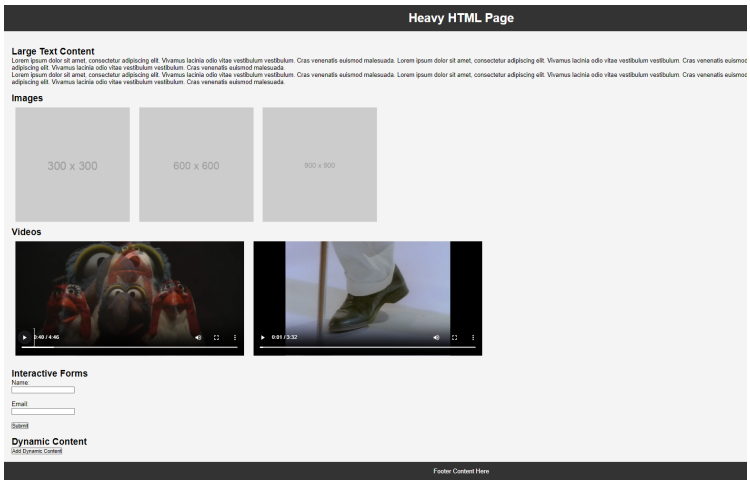
A6: Create a functional loading animation (Level 1)

Watch the video and recreate the spinner loading animation. The animation should only run while the provided HTML page is loading. The browser must be settled with a slow 3G connection and 6x slowdown CPU to test this.

Loading animation example:



Page displayed after animation:



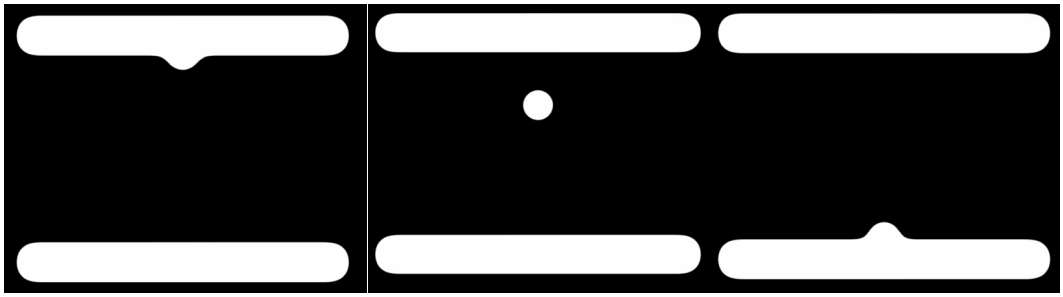
Requirement Notes

- The loading animation have been created
- The loading animation follows the provided media files
- The loading animation is displayed only while the page is loading

A13: Drip (Level 2)

No code is provided. Please implement a looping drip effect only using HTML and CSS, following the provided video.

These are some screenshots from the video.



Requirement Note

- Please refer to the reference video and create the same effect.

A22: CSS Bar Chart (Level 1)

Please implement a bar chart using only HTML and CSS.

At first, the screen will be blank, but when you hover the mouse over the chart area, the bars will rise sequentially from the bottom, starting from the left bar.

The values and colors of each data point are not important.

You can use html and css, but not javascript

Please refer to resource/demo.mp4

Requirement Notes

- The bar graph area is displayed and the initial screen is blank.
- When you place the mouse over the chart area, the bars move up in order from the bottom.
- When you place the mouse over the chart area, the bar goes up sequentially starting from the left.

Tasks for Front-end Development

B9: Image Compare(Level 2)

Given the HTML:

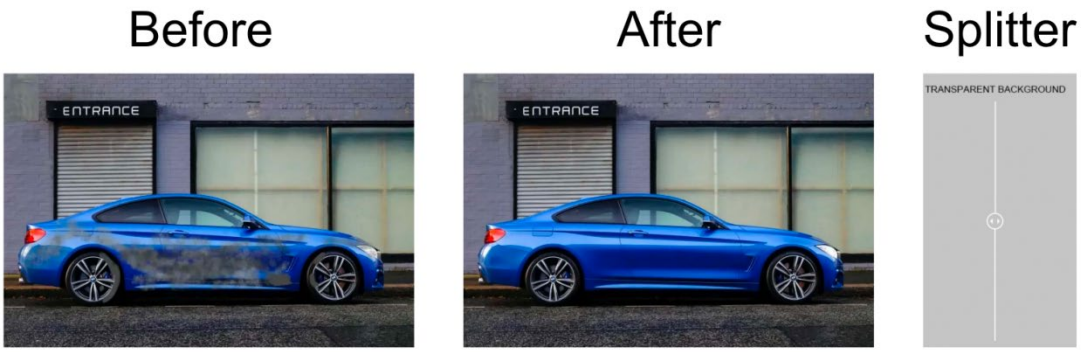
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Image Compare</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div>
        <input type="text">
        <img src="">
        <img src="">
    </div>
</body>
</html>
```

You need to use the provided index.html, style.css (empty file), two images (before.jpg and after.jpg) and a splitter.svg file to implement the Image Compare functionality. The Image Compare container has a splitter element. This element displays before.jpg on the left and after.jpg on the right. You can adjust the display ratio of the two images by clicking on the container or dragging the splitter element. As you adjust the ratio, one image will be displayed more prominently, while the other will be displayed less.

You cannot add or remove any HTML elements (except pseudo-elements). You can modify the attributes of existing HTML elements (including event listener attributes).

A sample video, "compare.mp4", is provided for reference. Below are some frames from the video:

The media files sample:

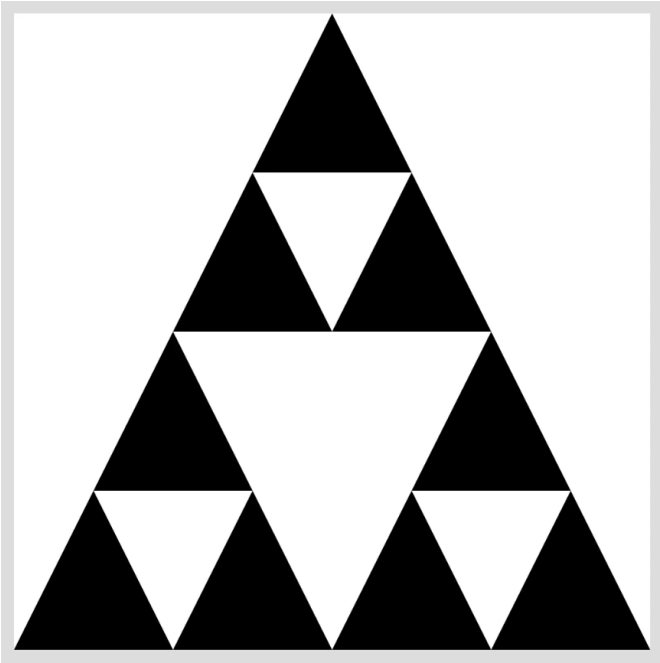


Requirement Notes

- The splitter left is before.jpg, the splitter right is after.jpg. No HTML elements were added or removed.
- The splitter can be moved by dragging with the mouse or clicking within the container.
- The image's effect is same as the compare.mp4 video.

B35: Fractal Triangle (Level 1)

After entering the number of iterations and pressing the button, you should draw the Fractal structure by repeating it the specified number of times.



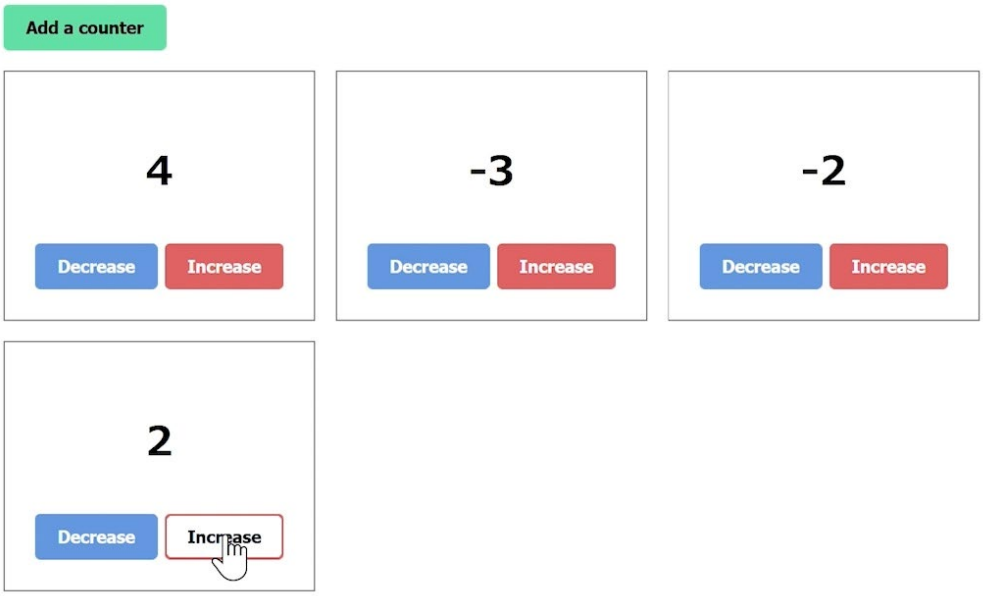
(Refer to demo.mp4)

Requirement Notes

- There is a canvas area, a form to input numbers, and a button.
- After entering the number of iterations and clicking the button, a fractal image is drawn.
- The drawn image has the same fractal structure as shown in the video (equilateral triangle structure).

B39: Multiple Counters (Level 1)

Using the provided video as a reference, complete the function using JavaScript.



Requirement Notes

- Can add a counter
- Can add multiple counters
- Each counter can independently increase and decrease
- UI is adjusted with user experience in mind

Tasks for Back-end Development

C1: Folder Zip (Level 1)

Please implement a simple web page for compressing folders. Users can select a folder (not empty) and click the "Compress" button to compress it into a zip file (The file name is the same as the uploaded folder name) and automatically download it. If the uploaded folder contains empty subfolders, those subfolders will not be included in the compressed archive. The folder for testing will be provided as media files

C9: API CORS (Level 2)

A Front-end application and a Back-end application are provided.

Access the Back-end through the VM Apache server, and use `npm run dev` for the Front-end application, making sure to use a different port than the Back-end.

When accessing the Front-end, 4 methods do not execute properly due to CORS.

Modify only the `cors.php` file in the Back-end application to ensure all requests execute correctly.

Run Front-end Application

You can run the development server by executing the `npm run dev` command.

Request address settings

You must open the front-end's `index.html` and modify the `API_HOST` variable to match your back-end.

(node_modules are provided in a compressed file. Be sure to refer to README.md)

Requirement Notes

- When you open the development server and connect, requests are sent to the back-end application.
- The GET method request works correctly.
- The POST method request works correctly.
- The PUT method request works correctly.
- The DELETE method request works correctly.

C11: Image Mosaic (Level 1)

You need to dynamically create a mosaic of the provided image.

The completed image should be made up of square cells, with each cell having the average color value of the corresponding area of the original image.

The size of the cells can be set using the query parameter `cell_size`. If `cell_size` is not provided, it should be set to 50px.

Example) When cell size is 50px:

Requirement Notes

- The mosaic is created according to the size specified by the `cell_size` query parameter.
- If the `cell_size` query parameter is not provided, the mosaic is created with cells of size 50px.
- The mosaic image is the same size as the original image.

Instructions to the Competitor

Each mini Test Project folder should be separated and self-contained.

Please include only exactly the required mini test project. And do not include any not related folders or not related mini Test Projects.

Other

This project will be assessed by using Google Chrome web browser.

Marking Summary

	Sub-Criteria	Marks
1	Mini Test Project General	4
2	Design Implementation	2
3	Front-end Development	2
4	Back-end Development	2