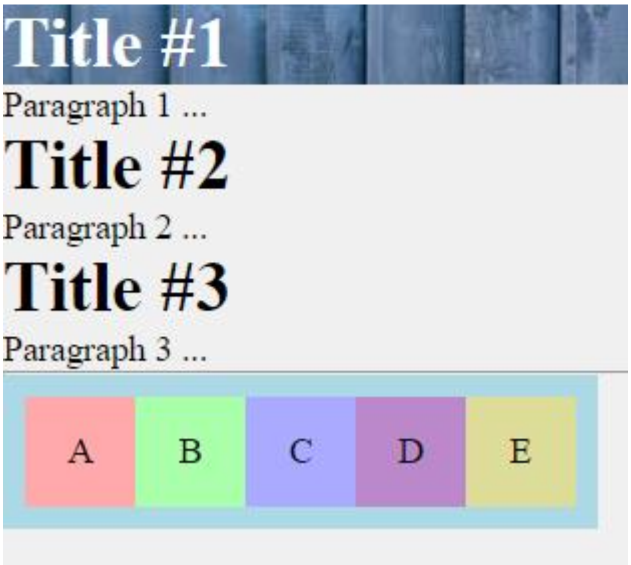
ECLT 5830 (Term 1 2020-21) Final Exam (Full mark 100%) 3 hours

Answer all FOUR questions.

1. [25%] HTML and CSS (All files related to this question are inside folder “q1”)



Title #1

Paragraph 1 ...

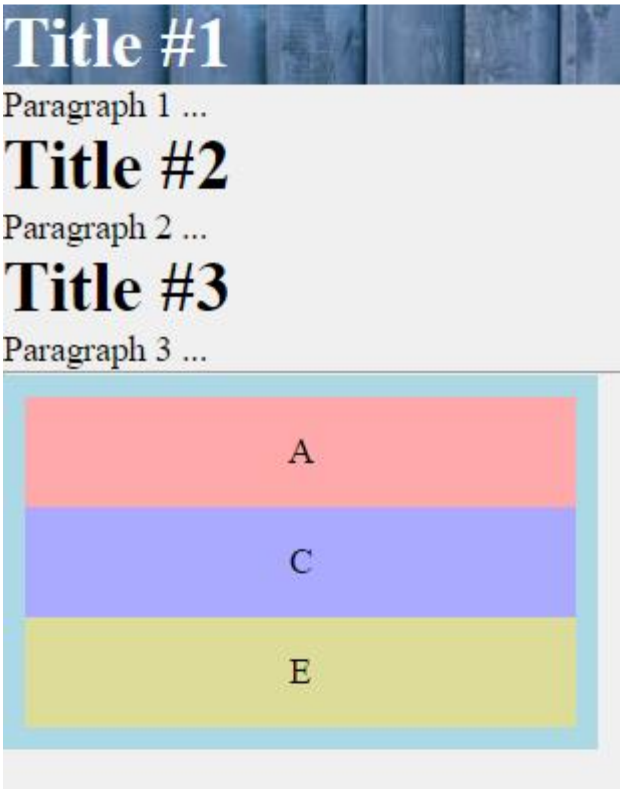
Title #2

Paragraph 2 ...

Title #3

Paragraph 3 ...

Fig. 1a: A sample wide-screen view



Title #1

Paragraph 1 ...

Title #2

Paragraph 2 ...

Title #3

Paragraph 3 ...

Fig. 1b: A sample narrow-screen view

Without modifying `q1.html`, edit only `q1.css` to produce a UI that meets the following requirements:

a) [5%] Apply the following styles ONLY to the **first** H1 element:

- Set the color of the text to white;
- Use the image “`bg.jpg`” as the background image.

In (b) and (c), the DIV element refers to the DIV element with `id=parent`, and SPAN elements refer to the SPAN elements inside the DIV element.

b) [10%] When the width of the viewport is more than 480 pixels:

- Set the width and height of the SPAN elements to 50 pixels;
- Leave no space between adjacent SPAN elements;
- Leave a gap of 10 pixels between the DIV element and SPAN elements;
- Center the text content of the SPAN element vertically and horizontally.

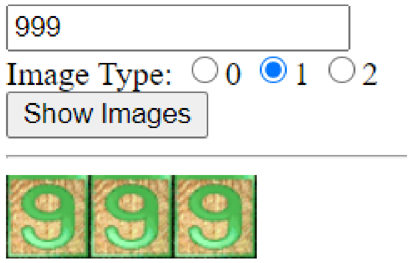
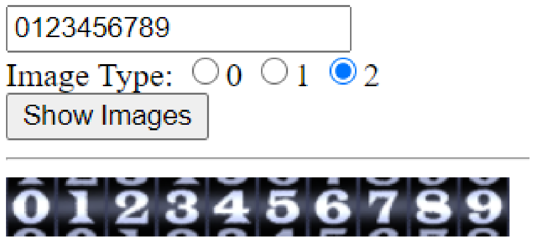
c) [10%] When the width of the viewport is less than or equal to 480 pixels:

- Set the height of the SPAN elements to 50 pixels;
- Set the width of the SPAN elements to 100% of its parent;
- Leave no space between adjacent SPAN elements;
- Leave a gap of 10 pixels between the DIV element and the SPAN elements;
  - Note: Because of this gap, the actual width of the SPAN elements will be 20 pixels less than the width of the DIV element.
- Center the text content of the SPAN element vertically and horizontally;
- Do not show the SPAN elements with `id=c2` and `id=c4`.

If your solution involves modifying `q1.html`, you can only get at most 15 points for this question.

2. **[25%] JavaScript DOM (All files related to this question are inside folder “q2”)**

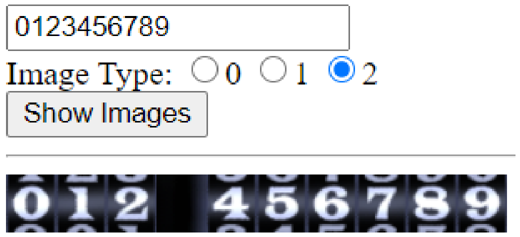
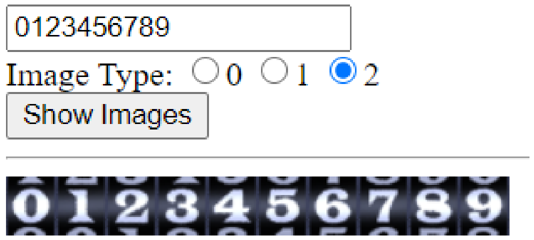
(a) [15%] Modify `q2.html` so that when a user clicks the “Show Images” button, the page will show the equivalent digit images (of the specified image type). The following are some sample results:

 999 Image Type: ○0 ●1 ○2 Show Images	 0123456789 Image Type: ○0 ○1 ●2 Show Images
Fig 2a: Sample 1.	**Fig 2b**: Sample 2.

The mapping between “Image Type” and image filenames are as follow:

Image Type	Image Filenames (for digits 0, 1, …, 9, and “blank”)
0	`img/0a.gif, img/1a.gif, …, img/9a.gif, img/blank_a.gif`
1	`img/0b.gif, img/1b.gif, …, img/9b.gif, img/blank_b.gif`
2	`img/0c.gif, img/1c.gif, …, img/9c.gif, img/blank_c.gif`

(b) [10%] Extend your solution in part (a) so that when the user clicks a digit image, the digit image will toggle between the digit image and the blank image of the same type. By clicking the digit/blank images one by one, the user should be able to toggle multiple digit images to blank images (and vice versa).

 0123456789 Image Type: ○0 ○1 ●2 Show Images	 0123456789 Image Type: ○0 ○1 ●2 Show Images
Fig. 2c: After the 4th digit image in the UI shown in Fig. 2b was clicked once (toggle from “3” to “blank”).	**Fig. 2d**: After the “blank” image in the UI shown in Fig. 2c was clicked once (toggle from “blank” to “3”).

Note:

- You do not need to validate the value in the input field; you may assume the value in the input field is always a sequence of digits.
- You may use any approach, library, and framework in your solution.
- “vue.js” is available inside the “q2” folder in case you need it.

3. [30%] Ajax, Express, Session (All files related to this question are inside folder "q3")

The folder "q3" contains an incomplete Node.js web application. When the web application is running, you can access the server from your browser via URLs in the form http://localhost:8080/...

Your objective is to complete the implementation of the web application according to the following requirements:

a) [5%]

```
GET /eclt5830/q3?sid=123456 HTTP/1.1
Host: localhost:8080
Origin: http://localhost:8080
```

Modify `index.js` so that when the server receives a HTTP request in the above format, the server will return the value of the following condition (as either "true" or "false") in the response body:

> The parameter "`sid`" exists in the request and its value is equal to the last 6 digits of your CUHK student ID.

b) [10%]

```
POST /eclt5830/q3 HTTP/1.1
Host: localhost:8080
Content-Length: 21
X-5830: Hello World!
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://localhost:8080
Cookie: c1=Alpha; c2=Beta; c3=Gamma

foo=123.875&b%3Dr=111
```

Modify `index.js` so that when the server receives a HTTP request in the above format, the server will return the following values in the response as **plain text** (not HTML) content.

- Value of the header field "`X-5830`"
- Value of the cookie named "`c2`"
- The sum of the two parameters in the body

For the request shown above, the content in the response should be:

```
Hello World!
Beta
234.875
```

Note: You may assume all fields marked `blue` in the request shown above always exists and have a valid value, and the values of the two parameters are valid numbers. Values marked `red` can vary.

c) [5%] Modify `index.js` so that the web application will support server-side session that meets the following requirements:

- The session should last for 30 minutes.
- The session should be available only to **GET** requests with path value begins with “`/abc/`”.
- The session should only be created when a client (without an active session) sends a GET request with path value begins with “`/abc/`”.

d) [10%]

Implement the function `send(value1, value2, value3)` in “`public/q3d.html`” so that the function will send the three parameter values to http://localhost:8080/q3d asynchronously. Each of the values can be a string, `null`, or `undefined`. Upon receiving the response from the server, the client-side JS code should use `console.log()` to output the response content.

Modify `index.js` so that when the server receives the request sent by the function `send()`, the server will return the “type” (as one of “`string`”, “`null`”, and “`undefined`”) of the three values in the response in the following format:

```
type of value 1, type of value 2, type of value 3
```

For example, when `send(undefined, "null”, null)` is called on the client side, the server should return the following content in the response:

```
undefined, string, null
```

Notes:

- Except for (a), you can assume all the data received by the server are valid.
- For this problem, you need to install all the necessary modules such as “express” and “express-session” yourself.
- Please remove the “node_modules” folder in the app folder before you submit your solution.
- “jquery.min.js” (version 3.5.1) is available inside the “public/” folder in case you need it.

4. **<u>[20%] Short Answer Questions</u>**

a) [10%] Which of the following Ajax requests will involve a “CORS preflight request” to the target server?

A	`$.ajax({   url: 'http://www.foo.net/a',`
	`            method: 'POST', data: { a: 123, b: 456 }`
	`});`
	`// The current page is originated from http://www.BAR.com`

B	`$.ajax({   url: 'http://www.foo.net/a',`
	`            headers: { 'x-custom’: 'hello' }`
	`});`
	`// The current page is originated from http://www.BAR.com`

C	`$.ajax({   url: 'http://www.foo.net/a',`
	`            method: 'PUT', data: { a: 123, b: 456 },`
	`});`
	`// The current page is originated from http://www.foo.net`

D	`$.ajax({   url: 'http://www.foo.net/a',`
	`            method: 'POST', data: '{"a": 123, "b": 456}',`
	`            contentType: 'application/json'`
	`});`
	`// The current page is originated from http://www.BAR.com`

E	`$.ajax({   url: 'http://www.foo.net/a',`
	`            method: 'POST', data: { a: 123, b: 456 },`
	`});`
	`// The current page is originated from http://foo.net`

b) [10%] A blogging website has the following characteristics

- Only the owner of the website can create/update/delete blogs.
- A visitor can interact with a blog article by giving star rating to the article or by sharing the article at some social network website.
- The website has no CORS vulnerabilities.

Can this blogging website become vulnerable to XSS attack? Explain.

What to turn in?

a) Prepare a ZIP file (not ARJ or other archive formats) containing
- The folders `q1`, `q2`, `q3` (with your solutions)
- The document (.txt, .doc, .docx, or .pdf) with your answers to question 4

b) Upload the ZIP file to blackboard (I will mainly assess the solution submitted to Blackboard).

c) As a backup plan, please also send the ZIP file as email attachment to chengjiunyuan@cuhk.edu.hk with the keyword "FINALEXAM" in the email subject.

d) Please double check your submitted ZIP file to ensure it contains all your solutions.

-------- End of Question Paper --------