

**CS2204 Fundamentals of Internet Application Development**  
**Course Work No. 2 (CW2)**  
**Semester B 2024-2025**

**13% Marks**

**Due Date: 18 April 2025, Friday, 23:59 PM [Week 13]**

**Learning Outcomes:**

- Able to arrange the positions of web elements
- Apply CSS3 techniques
- Write JavaScript and execute them in Web pages
- Use JavaScript to dynamically manipulate HTML elements

**Overview**

The web pages you created in CW1 will be used. Enhance their interactivities and make all specified features work. CSS3 and JavaScript will be added, and you could also change the HTML/CSS in CW1, if necessary to facilitate the functioning of JavaScript. Detail requirements are listed in the following sections.

Guideline for late submissions:

- Late within **6 hours** - **10% marks deduction**
- Late within **6 ~ 24 hours** - **50% marks deduction**
- After **24 hours** - **no** submission will be accepted

We do **NOT** grant any extension, unless you have a solid justification (e.g., medical issues) with supporting documents.

**Submission**

- Submit a zip file of your website to Canvas with appropriate folders set up so that it could be test directly by unzip - be careful with your URLs; they should work when your submission is test as local files in other computers or as Web pages served by a Web server
- Image sources needed in this assignment are on Canvas (same as CW1)
- Please DO NOT include videos in your submission. Instead, they should be directly linked to: <https://personal.cs.cityu.edu.hk/~cs2204/video/> (Note that this link can only be accessed within CityU network. If you are outside campus, you should use VPN). **Videos for the promotion information block:** Castle.mp4, Castle.ogg
- DO NOT use YouTube video; should use any one given video link in the server folder.

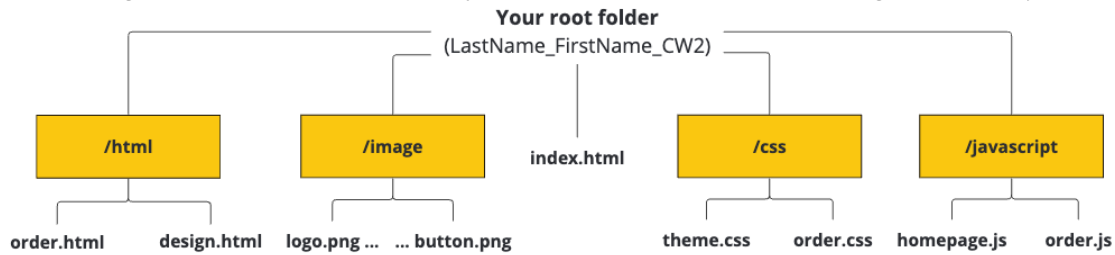
**Assessment criteria**

These assessment criteria will also be found under the Canvas submission page “Rubric”

Criteria	Description	Marks
Web elements positioning with HTML and CSS	Appropriately arrange the positions of web elements	2
CSS3 effects	Implement required effects with CSS3	2
JavaScript basics	Appropriately link and execute JavaScript	2
JavaScript features	Implement required features with JavaScript	7
Good web dev practice	Consistency, code readability, user input validation	/

## 1. Requirements of Structure

The website structure is the same as what specified in CW1. In particular, newly added css files should go under “/css”, and newly added JavaScript files should go under “/javascript”

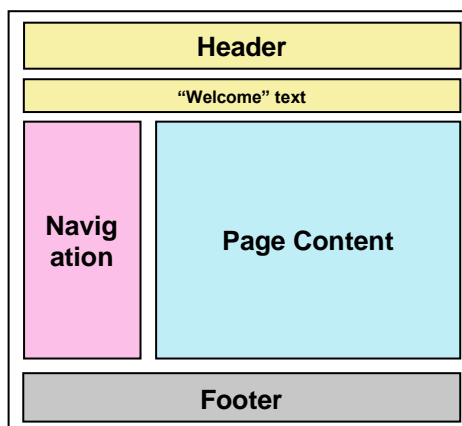


In building this website, **all contents unless specified could be made up by you.** If you use information and ideas of other people from the Internet, you **MUST** acknowledge and cite the source in “design.html”

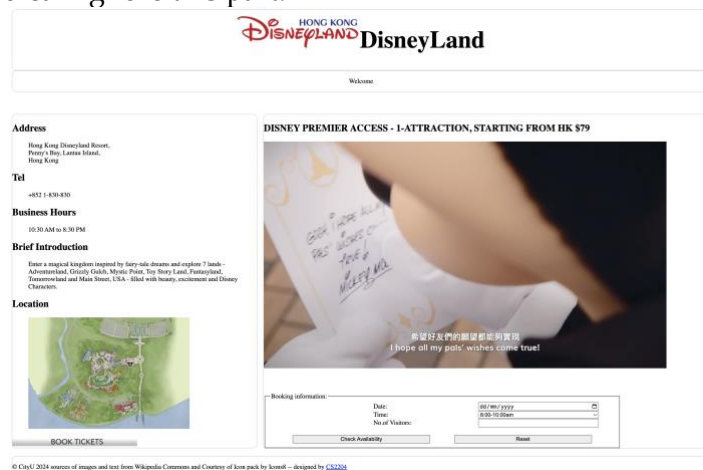
This assignment covers what we learned from Lec 05 to Lec 09 as well as Lab 04 to Lab 08. You are required to apply **CSS** and **JavaScript** to make this website fully functional. Thus, you should be careful about potential interaction issues, such as user input does not meet requirements and any potential JavaScript bugs.

## 2. Home page web elements positioning

You need to achieve a web layout similar to what we did in Lab4, including a header, navigation, content, and footer sections. If you have already implemented the same positioning requirements in CW1, you can ignore this part.



(a): Illustration of layout



(b) Example of the finished homepage

**2.1** You should implement the positioning in “theme.css”. To make it clear for us to see the positions of each section, please add **block borders** accordingly. The style of the borders is not specified, so you can choose any color, form, and thickness based on your preferences.

**2.2 header section.** The logo and Web site title sit in the same line and are centered together as one unit. The “Welcome” is in another block, which is centered horizontally.



**2.3 navigation section.** This section is in left column, including address, tel number, hours, brief intro, location info, and the option for “Book Tickets.”

**2.4 content section.** This section is in the right column and contains two blocks (vertically arranged). The top block shows the video, and the bottom block shows the booking information

- top block: the video should be centered horizontally and clearly visible within the promotion information block.



- Bottom block: It contains labels, inputs, and buttons in the booking form should be aligned properly as follows

Booking information:

Date:	<input type="text" value="dd/mm/yyyy"/>
Time:	<input type="text" value="8:00-10:00am"/>
No.of Visitors:	<input type="text"/>

**2.5 footer section.** It starts from a new line at the bottom of the page.

### 3. Order Page Web elements positioning and effects

This page contains header, footer, menu and ordered items four major parts as follows.



**3.1** Create a css file, named “order.css” to enable the following effects.

**3.2** For “Ordered Items”, the whole block occupies 22% of the browser window width. Text in the description column is aligned left and all quantity text are aligned right. For other styles, you can take the following figure as an example for reference. The exact fonts and line spaces are not required, and you can select appropriate values for them.

**3.3** Set the 3 menu item headings (i.e., “Tickets”, “Dinings” & “Hotels”) to equal width and occupy the whole width of the menu block

Menu		
Tickets	Dinings	Hotels

3.4 When mouse moves over each menu, menu items are shown as follows. When mouse leaves the menu, the menu items disappear, and the menu block gets back to its original format as shown in the figure above. **WITH CSS ONLY (i.e., no JavaScript)**

- When mouse is on the "Tickets" menu, three menu items are as shown as follows:

Menu		
Tickets	Dinings	Hotels
 1-Day Ticket <input type="text"/> Add	 1-Day Ticket (Child) <input type="text"/> Add	 2-Day Ticket <input type="text"/> Add

- When mouse is on the "Dinings" menu, three menu items are as shown as follows:

Menu		
Tickets	Dinings	Hotels
 Explorer's Semi Buffet <input type="text"/> Add	 3-in-1 Park Meal Voucher <input type="text"/> Add	

- When mouse is on the "Hotels" menu, four menu items are as shown as follows, where the first three items in the first row with equal width and the fourth item starts from a new line.

Menu		
Tickets	Dinings	Hotels
 Standard Room <input type="text"/> Add	 Deluxe Room <input type="text"/> Add	 Sea View Room <input type="text"/> Add
 Kingdom Club Room <input type="text"/> Add		

#### Hints for 3.4:

- When the headings are pointed by mouse cursor: the heading background color is set to white (or the background color of the menu); the corresponding group of menu items are displayed while the other groups are hidden; note the menu items will not go away as long as the heading or any menu item is being mouse-over
- If the cursor moves away from any heading and menu items, the display goes back to the initial stage. The technique is to set CSS property of one element by mouse-over another element; need to think in terms of using a selector (relationship) for the menu item group that involves the corresponding heading and then apply the hover pseudo class to the heading.

## 4. Home page interaction

Create a JavaScript file "homepage.js" under "javascript" folder and write all your homepage-related scripts here. Do not forget to link the script to index.html.

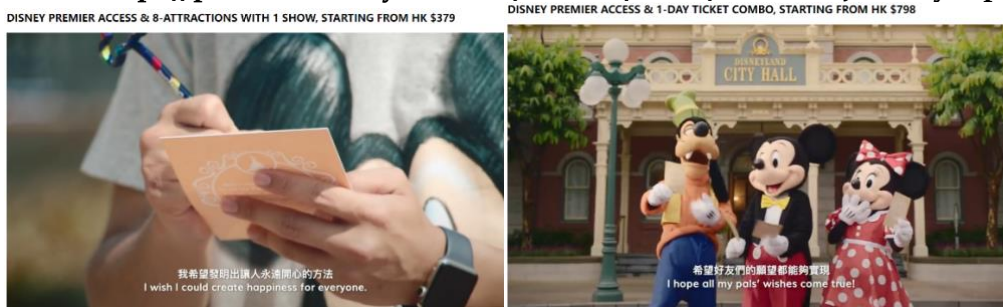
4.1 Declare an array to hold 3 promotion text messages.

- "DISNEY PREMIER ACCESS & 1-DAY TICKET COMBO, STARTING FROM HK \$798"
- "DISNEY PREMIER ACCESS & 8-ATTRACTIONS WITH 1 SHOW, STARTING FROM HK \$379"
- "DISNEY PREMIER ACCESS - 1-ATTRACTION, STARTING FROM HK \$79"

When the page is loaded, pick a message at random and display it in the promotion information block. Hints: if your array's name is message, first randomly generate a number from 0 to 2, so that you can randomly select from message[0] to message[2] to display as .innerHTML

4.2 (continuing from 4.1) Display the next message after 3 seconds and keep on cycling through the messages continuously every 3 seconds. Hints: call `setInterval()` to execute a function that changes the message every 3 seconds (3000 milliseconds).

4.3 Use one of the two videos below as a default in your HTML. At the end of showing one video, switch to another and repeat this continuously. Single source mp4 is good enough  
<https://personal.cs.cityu.edu.hk/~cs2204/video/Castle.mp4>  
[https://personal.cs.cityu.edu.hk/~cs2204/video/Musical\\_Journey.mp4](https://personal.cs.cityu.edu.hk/~cs2204/video/Musical_Journey.mp4)



4.4 When the Reservation form is submitted, check that all fields are not blank (i.e., empty string or all spaces). Display an error message “Data not completed, please re-enter” above the *Submit* and *Reset* buttons when error is found. **Do not use `alert()`**

Hint: In the form, you can create this error message which by default is hidden or not displayed (using CSS to set the style). Then when you detect the form is incomplete after the “Check Availability” button is clicked, you can change its display style. For example, if the id of this message is “warning”, the following code can change its visibility

```
document.querySelector("#warning").style.display="block"; //display the message
document.querySelector("#warning").style.display="none"; //hide the message
```

Booking information:

Date:

Time:

No.of Visitors:

dd/mm/yyyy

8:00-10:00am

Data not completed, please re-enter.

Check Availability

Reset

4.5 If there is no error in the form data, call the function `reserve` in the library as `reserve(date, time, no-of-visitors)`. Display the message “Reservation done. Thank you.” **in an alert window** or “Disneyland has reached the maximum number of visitors for the day” for getting a function return of `true` or `false` respectively. Cancel the form submission in all situations after message display.

Here, we provided the external js function `reserve` for you, which can be found at: <https://personal.cs.cityu.edu.hk/~cs2204/cs2204cw3.js>. You should **link this file to the home page** instead of making a copy to your “homepage.js file”.

## 5. Order Page Interaction

In “javascript” folder, create “order.js” and write all your scripts related to the order page there. Do not forget to link the script to order.html



5.1 After page loading, show the *Tickets* menu items and set the background color of the Tickets heading to white and the other 2 headings' with a color of your choice

5.1 Set up event handlers for all "Add" buttons in the menu items. You should assume the number of buttons/menu items is only known after page load. Write codes for the event handler such that when a button is clicked, a row is added at the end of ordered items in the table. You may modify HTML of the ordered items table if needed. **Merging of ordered items with the same description is NOT required.**

Hints:

- Consider using *ids* in Add buttons to identify the menu item code and to relate to the quantity input boxes as well as images, e.g. ,s1, s1-qty and s1-img
- Description of the item can then be obtained from *alt* or *title* attribute of the image. Use the clicked Add button's id to build strings to select the qty input box and the image.

5.2 Write a function *recal()* to calculate the total quantity whenever the table of ordered items is updated with a new or deleted entry. The new total quantity should be put in the table footer row.

Ordered Items	
7 Jul 2022 22:22	
Table:4-No.of Guests 3	
Description	Qty
1-Day Ticket	1
3-in-1 Park Meal Voucher	1
Standard Room	2
Deluxe Room	1
Total	5
undo	

5.3 Set up and write the *onclick* event handler for the *undo* link. Delete the last ordered item for every clicking of the *undo* link and update total quantity. **Remember to cancel the onclick event at the end of the event handler** (i.e., clicking the link will not go to any destination, not even an internal link). **Need to handle the case of empty table, with only header and footer.**

## 6. Free Design with CSS3

Use **at least two different** CSS3 techniques we covered in the lecture in your web pages, including but not limited to gradient colors, transitions, animations. There is no specific requirements regarding which techniques to use, but **you should list what you used in the Design page ("design.html")**.

## 7. Assessment

7.1 All the requirements from section 1 – 6 are met.

7.2 User input validation: make sure you in the places where users need to enter information, the website should validate their entered information before processing the them. Hint: you can use new input types such as numbers to make sure that user input is number

7.3 All the .css and .js files are correctly linked

7.4 Your code should be well organized and readable

7.5 Use of other JavaScript libraries or frameworks such as jQuery, Angular or React.js, etc. is **NOT allowed**

~ End ~