

COMPX222 – Assignment 2:

JavaScript Form Validation and DOM Manipulation

(30%)

Deadline: Friday 10th May by 23:59 (at the latest)

Task

In this assignment, you will build a booking form for a hotel or a similar lodging business. Using the booking form, users will be able to enter information about the rooms they want to book, as well as their payment details. You will use JavaScript to validate the form and check that all the information entered by users is correct. If all the information the user submits is valid, your page should display a JavaScript `alert ()` ; with the message: “Booking Successful!”

The goal is to demonstrate your ability to use JavaScript (for DOM manipulation and form validation), as well as HTML and CSS (for creating usable, well-structured forms), so you must not use any external libraries or templates.

You are free to imagine any type of hotel you want as well as the kind of accommodation they offer. However, the form must meet all of the requirements listed in the requirements section.

Note: security concerns, such as protecting sensitive data like credit card numbers and backend validation for user input, are beyond the scope of this assignment.

Requirements

1. Your booking form must be a **single HTML page**.
2. The form should have a **Booking Details section** containing:
 - a. An input to select the number of rooms. There should also be the option to select at least two different room types (e.g., standard room, double room, or deluxe room) and each room type should have a different price. You should use the number type input for these.
 - b. An input to select number of adult and child guests. You should use the number type input for these.
 - c. A date input to select the arrival and departure dates. You should use the date type input for this.
 - d. A **Calculate button** for calculating the total cost of the booking.
3. The form should also have a **Payment Information section** containing:
 - a. An input for selecting the users title (e.g., Mr, Mrs). You should use a select dropdown or radio input for this.
 - b. An input for the user’s first name.
 - c. An input for the user’s last name.

- d. An input for the user's email.
 - e. An input for the user's phone number.
 - f. An input for selecting the Card Type. The options for this should be: Union Pay, MasterCard, and Visa. You should use a radio input for this and display the logo for each card type.
 - g. An input for the card holder's name.
 - h. An input for selecting the card expiry date (the month and year that the card expires).
 - i. An input for selecting the card CVV (the 3-digit code on the back of the card).
 - j. A **Cancel button** for clearing the form.
 - k. A **Pay Now button** for submitting the form.
4. The following **validation should be applied to each field**:
 - a. Arrival and departure dates: cannot be empty.
 - b. Rooms (e.g., standard, double): must be a number and must be greater than 1.
 - c. Guests (e.g., adults, children): must be a number and must be greater than 1.
 - d. Title, first name, last name, card holder name: cannot be empty.
 - e. Email: cannot be empty and must be a valid email address. You can use regex to check this is a valid email address.
 - f. CVV: cannot be empty and must be a 3-digit number.
 - g. Card expiration: cannot be empty and must be and year and month.
 - h. Phone number: cannot be empty and must be an 11-digit number.
 - i. Card type: one of the radio options must be selected.
 - j. Card number: cannot be empty and must be a 16-digit number.
 5. When the **Calculate button is pressed**, the form should calculate the total cost of the booking based on the number of days and rooms. For example, if the user wants a room of 400 RMB/night for 3 nights, the total should be $400 * 3 = 1200$ RMB. If all of the booking information (arrival, departure, rooms, number of guests) has been filled in, the calculated total should be displayed to the user. In any of these fields have not been filled in, the form will show the appropriate validation telling the user to fill in the missing fields.
 6. The **Payment Information section should be hidden by default**. It should only appear when the Booking Details section has been properly completed and the total cost has been displayed.
 7. When the **Cancel button is pressed** all form data should be cleared and the Payment Information section should be hidden.
 8. When the **Pay Now button is pressed** the form should use a JavaScript alert(); to display the message "Booking Successful!" if all of the inputs in both sections are correctly filled in. If any of the inputs are empty or incorrectly filled in, form will show the appropriate validation telling the user to fill in the missing fields.
 9. The booking form **data must be validated with JavaScript (in booking.js)** and invalid data should be reported with appropriate error messages, styled to be recognizable as errors
 10. The validation must occur both (1) when the user moves between input elements (when the input element loses focus) and (2) when the user clicks on the final " Pay Now" button.

11. The HTML form elements should be appropriate for the type of data that they capture, e.g., type="date" for date inputs and type="number" for number inputs.
12. Your webpage **must be responsive**. This means it should be usable on different screen sizes and devices. You should use CSS grid, flexbox and media queries to achieve this.
13. You must add comments to describe the main parts of your JavaScript, HTML and CSS code.

Restrictions

The purpose of this assignment is to demonstrate your ability to use JavaScript for form validation and DOM manipulation. Therefore, the following restrictions apply:

- **DO NOT** copy and paste large blocks of JavaScript, HTML or CSS from other sources.
- **DO NOT** use visual design tools (such as Dreamweaver) that generate the HTML/CSS code.
- **DO NOT** use external libraries or frameworks (such as Bootstrap or templates).
- **DO NOT** use third-party JavaScript tools, frameworks, or libraries.
- **DO NOT** use the HTML required attribute as this could prevent some of your JavaScript validation being triggered.

Guidelines

Project Organisation

Organize your project properly. In the root folder of your website, you will have a file called index.html, a file called style.css (for your separate CSS code), and folder called img for your images (and videos). Your folder structure should look like this:

```
/root
  /css
    style.css
  /img
    image1.jpg
    ...
  /js
    booking.js
booking.html
```

Filenames (including images) should only contain ASCII characters, so you should avoid using Chinese characters. This is to prevent problems when zipping/unzipping.

Images

Do not use absolute filenames for image and videos because if you do, they will only work on your computer. All links to images and videos should use relative filenames.

Correct (relative):

```
src="img/image1.jpg"
```

Incorrect (absolute):

```
src="/Users/alex/Documents/your-website/img/image1.jpg"  
src="C:\your-website\img\image1.jpg"
```

All **images should be appropriately sized for the web** and be no larger than they need to be. Large image files can significantly impact page load times, leading to a poor browsing experience for users.

CSS

You must use an external CSS file (style.css) for all of your CSS code then include it in your HTML file using a <link> tag inside the <head>. Do not include any CSS code in the HTML file (either inline in the HTML elements or in style tags at the top of the HTML page).

Correct:

```
<head>  
  <link rel="stylesheet" href="css/style.css">  
</head>
```

Incorrect:

```
<p style="color: green">  
  
<style> ... </style>
```

HTML

ALL text must be enclosed in the correct HTML tags. Do not leave any un-tagged text in the HTML file. All HTML must be valid HTML5 and you should use semantic HTML5 elements where appropriate.

JavaScript

Keep your **JavaScript code separate from your HTML file**. Create an external JavaScript file named booking.js and include it in your HTML file using the <script> tag just before the closing </body> tag:

```
<body>  
  <!-- HTML content here -->  
  <script src="js/booking.js"></script>  
</body >
```

Attach event listeners in your JavaScript file rather than using inline event handlers in your HTML. This ensures a cleaner separation of behaviour from structure.

Correct:

```
document.getElementById(calculateButton).addEventListener('click',  
calculateTotal);
```

Incorrect:

```
<button onclick="calculateTotal ()">Calculate</button>
```

What to Submit

All your material for this assignment must be submitted using Blackboard.

1. Create a folder with a name of "320XXXX-Assignment2," where 320XXXX is your student ID. You should end up with a single ZIP file called **320XXXX-Assignment2.zip** containing your website files inside (see the Project Organisation section for details). Use only ASCII characters for filenames (no Chinese characters).
2. Submit the ZIP archive via Blackboard. On the course page you will find a section 'Assignments' with a link entitled 'Assignment 2 Submission' (or similar). Follow the link and upload your ZIP file.

No other mechanism for submission will be accepted.

NOTE: your site should work 'as is' when unzipped from your ZIP archive, without any modification. You should check that this is the case by unzipping your archive and testing the site yourself.

Assessment

Your mark for this assignment contributes to **30% of your overall grade**. Zip your website up into a single file and submit it on Blackboard before the due date. The marking will be as follows:

HTML Markup Including structure of the HTML document, appropriate form elements, attributes and input types	20%
CSS Styling Including overall style and consistency, usability, responsiveness, and use of selectors	20%
JavaScript Including validation, displaying or error messages, total calculation, elements displayed/hidden as required, use of functions and appropriate techniques	50%
Project Structure Including file and folder organisation, file naming, external CSS and JavaScript	10%
Total	100%

You will lose marks if you break any of the restrictions or guidelines.

The assignment is **INDIVIDUAL work**. You are not allowed to work with other students or copy code from other sources and you must not share your code with any other student. If any assignment description is not clear to you, please ask your teachers.