COMPX222 – Web Development

Assignment 2: Form Validation and Server-Side Scripting (15%)

Due on: Friday 12nd May at 8pm

Submission Instructions: Submit your files on Blackboard in a ZIP folder called 320XXXX-Assignment2.zip where 320XXXX is your student ID.

What to Submit: Website files (booking.html, confirmation.php, "js" folder, "css" folder, "images" folder) in a ZIP folder called 320XXXX-Assignment2.zip where 320XXXX is your student ID.

Goal: To apply knowledge of HTML mark up, CSS styling, DOM manipulation, JavaScript in the development of a small web site, including forms for data input by the user, and apply knowledge of PHP to develop server-side script to process user input.

Overview:

In this assignment we will build a *booking form* for hotel rooms and breakfasts or similar "lodging business".

You have the freedom to imagine their name and where they are, as well as the kind of accommodation they offer, but there should be **3 types of rooms with different prices**.

Using the booking form, a user can request to book a room by giving out their personal information and the data on what they want to book. They also need to provide credit card information to secure the booking.

If this information is valid and the form is submitted, there should appear a *confirmation page* displaying all the data that was submitted.

Task

Step 1: booking form (booking.html, css/booking.css and images folders).

You need to make one HTML page containing the booking form. Give it a simple and clean look:

- A Header on top with an image or logo and the name of the business,
- A booking form taking most of the page,
- A footer on the bottom with copyright and contact information.

You don't need to make the page responsive, but you will need to create a CSS file to style the booking form page (e.g., considering fonts, colors, sizes, and positioning). All the images should be placed inside the "**images**" folder, and CSS style file (**booking.css**) in the "**css**" folder.

The booking form should contain the following elements:

1. A **personal data** fieldset contains:
   - Title - three options: Ms. , Mr. and Mrs.
   - User’s first name
   - User’s last name
   - User’s email address (use *email* type)
   - User’s phone number (use *tel* type)

2. A **room information** fieldset contains:
   - Room type (such as standard room, double room, or deluxe room) and the number of each room type that the user is booking; there should be a maximum for each room type (e.g., at most 10 standard rooms).
   - Number of adults and number of children.
   - Start and end dates of the reservation period (use *date* type), and Estimated Arrival Time (use *time* type).
   - Breakfast (*checkbox*) – if user wants to include breakfast. The breakfast price per night is different for adults and children.
   - A total price calculation – each type of room should have a different price. If breakfast is checked, the breakfast cost per person is added to the total, multiplied by each night.

   For example, if the user wants a room of 400 RMB/night for 3 nights (2 adults 1 child) and the breakfast – which costs 50RMB per adult and 20RMB per child – the total should be (400+50*2+20*1)*3=1560RMB . Note that this is just an example, you can set your own prices.

3. A **payment** fieldset – this should be shown only when the above two fieldsets have been completed properly – contain:
   - Card Type (use *radio button*). This option should have (1) UnionPay (2) Visa, (3) MasterCard– it's recommended to use their logos as labels.
   - Card Number. Each card type has its own numbering rule.
     (1) UnionPay cards have 16 to 19 digits and start with 6.
     (2) Visa cards have 16 digits and start with 4.
     (3) MasterCard cards have 16 digits and start with 5.
     You should validate the starting digit number and the total number of digits, according to the chosen card type. Note that you could use *regular expressions* to validate the card number but don't need to implement too complex validations (e.g., checksum).
   - CVV. This is the secret code on the back of the card. It must be of 3 digits.
   - Cardholder Name. The name of the owner as written on the card.
   - Card Expiration (use *month* type). It is the month and year of the card expiration date. If the card is expired an error message should appear.

4. **Two buttons** – "*Confirm and pay*" to submit the form and confirm the payment, and "*Cancel*" to cancel and clear the form data.

Important:

- The HTML elements should be appropriate for the type of data that they capture.
- Elements should be displayed/hidden when appropriate.
- You need to style the form using appropriate CSS styles. You do not need to achieve modern or elegant design but give inputs appropriate and consistent size and positioning. All CSS styles should go to the CSS file (css/booking.css).

Step 2: Form Validation (js/booking.js).

You must make sure that booking form data is validated with JavaScript (booking.js) and invalid data is reported with appropriate error messages, styled to be recognizable as errors. The JavaScript (booking.js) should be placed under "**js**" folder (**js/booking.js**).

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 "Confirm and Pay" button.
If all the booking and payment data is valid, it should be submitted with a **POST** request to a PHP script (confirmation page) when the user clicks on the submit button.

Step 3: Confirmation Page (confirmation.php).

The confirmation page (confirmation.php) on the server side should receive the data and display them in a HTML page.

(1) First, there should be a greeting and confirmation of the payment: "*Thank you, Mr. John Smith, we have received your payment order and it will be processed shortly. Here are the details of your order*". Of course, you should replace "*Mr.*", "*John*" and "*Smith*" with the data received from the user (title, first name and last name).

(2) After the greeting, there should be an **HTML table** displaying the data received from the user.

- Each row of the table should be one of the inputs from the user: the left column should have its label and the right column its value. For example, the first row should be | Title | Mr. |, the second row should be | First Name | John | and so on…

Element Name	**Form Data**
Title	Mr.
First name	John
Last name	Smith
Email Address	mweng@waikato.ac.nz

……

This includes the data on the room and on the payment. There must be also an additional field showing the total that will be paid.

- The credit card number should be masked with * and only the last 4 digits should be visible (for example, if the card number is 4123412341235678 then your table should display ************5678).
- The CVV number should be completely masked as ***.

The confirmation page should have the same header and footer as the booking form page and should be styled appropriately with CSS.

NOTE ON SECURITY: In a real live website you would NEVER send sensitive data such as credit card numbers without additional protection over HTTP. The HTTPS protocol will be discussed later in the lectures on web security, we will not use it in assignments. Also, to receive and process credit card information in the real world you would need to comply with very strict regulations and standards regarding the security of your networks, data storage and company procedures - for more information look up "PCI-DSS", i.e., Payment Card Industry Data Security Standard. But **these security concerns are not part of this assignment**.

Other requirements:

- **CSS**

Use a single CSS file (css/booking.css) to control the presentation and layout of your site's pages. The HTML page should avoid inline CSS and internal style attributes. The CSS file should not contain unused or repetitive CSS styles.

- **Comments**

Include appropriate comments in all your HTML, CSS, JavaScript, and PHP files.

- **PHP**

The PHP page (confirmation.php) should have a .php extension. In this PHP page, most of page will be HTML mark-up code and the PHP code will be placed between `<?php` and `?>` tags.

- **Validation**

All HTML must validate as HTML5, and all CSS must validate as CSS3 using:
HTML validator: https://validator.w3.org/nu/
CSS validator: https://jigsaw.w3.org/css-validator/

- **Third party libraries**

You are not allowed to use third-party JS tool, framework, or library.
If you include third party content in your site (such as images, icons, background, etc), please download the images to "images" folder and use the images of your own copy. Do not link to the original location of the image on the web.

- **Folder Structure**

Organize your resources by using folders:

(1) The HTML and PHP pages (**booking.html** and **confirmation.php**) should be in the root folder.
(2) The CSS files (**booking.css**) should be in the "**css**" folder.
(3) The JavaScript file (**booking.js**) should be in the "**js**" folder.
(4) All the images should be in the "**images**" folder.

Filenames (including images) should contain only ASCII characters to avoid problems when zipping/unzipping folders.

- **Strategy**

The server-side portion of this assignment hasn't yet been fully covered in the lectures. However, you can start creating the HTML page, then the form and working on JavaScript validation. As we learn about PHP in the class and you become more familiar with PHP, you can start adding the server-side component of the assignment.

<u>What to submit:</u>

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. Please use only ASCII characters for filenames! Place all files and folders for your web site inside that folder. Create a ZIP archive of that folder so that you end up with a file **320XXXX-Assignment2.zip**
(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 15% to your overall grade for this paper. Zip your website up into a single file and submit it on Blackboard before the due date. The marking will be as follows.

Appropriate folder structure and file names	5%
Overall look, style, and consistency	5%
HTML quality and validation – indentation, use of tags, appropriate attributes, comments	10%
CSS quality and validation – indentation, clarity, comments, use of selectors	15%
Appropriate elements and attributes in the form (booking.html)	10%
Form elements are displayed/hidden as required	5%
JS total calculation	5%
JS validation of the forms - includes validation as the user moves between controls and validation on submission, error messages	20%
JS overall quality – indentation, clarity, comments, use of functions, appropriate use of techniques and constructs seen in lessons	10%
PHP confirmation page – includes all required elements	5%
PHP script – absence of errors/warnings, indentation, clarity, comments	10%
Total	100%

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 of this information is not clear to you, please ask your teacher.