worldskillsuk

WSUK – National Finals Competition

Web Development

2025

MODULE D

SPEED

Val Adamescu
Lewis Newton
Mark Kiss

Contents

Introduction

This module will test your ability to apply your HTML, CSS, JS/PHP knowledge effectively and creatively. You will be required to complete several mini-test projects within 3 hours. You need to submit it before the time runs out. No additional time will be given for submission. There will be both easy and routine tasks, as well as more complex ones. For each mini–test project, there are three levels.

- *Level 1 (Easy – between 5 and 10 min to complete it)*
- *Level 2 (Medium – between 10 and 20 min to complete)*
- *Level 3 (Difficult – between 20 and 30 min to complete)*

Some tasks can be solved using either JS or PHP – provide only one solution.

Each mini-test project folder must be separated and self-contained. Inside the dist folder is an index.html that will automatically create a link for each task you have attempted/completed. Submit ONLY the attempted tasks (e.g. d1, d3, d7) not all the tasks.

NOTE: *The specific submission/upload requirements of the test project are provided separately.*

**Please always check the next page if the task has no continuity on the following page as content or samples.*

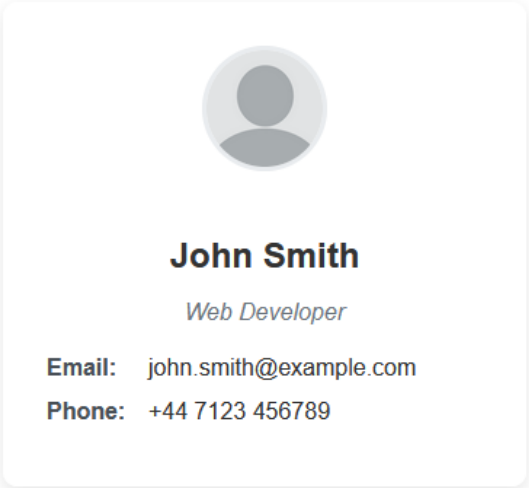
TASKS

★ D1 - Contact Card Display (L1)

Using CSS only (no JavaScript allowed), create a contact card positioned in the centre of the page. The provided CSS should not be modified above the designated area. A placeholder avatar image is provided in the ***`img`*** folder. You should not manipulate this image through imagine manipulation software (e.g. Photoshop), but instead through CSS. Do not use the provided sample image as your solution.

Requirements:

- Card width: 300px, centred on page
- Profile image: circular, 80px diameter
- Card: white background, border-radius (10px), padding (30px), shadow
- Texts:
  - Name "John Smith"
  - Title "Web Developer"
  - Email "john.smith@example.com"
  - Phone "+44 7123 456789"
- Hover effect: card moves up 5px



★ D2 - Text Content Analyser (L3)

Using JavaScript or PHP, implement a text content analyser that calculates various statistics and metrics from user-provided text. You should not modify any of the existing code.

PHP: Place your server-side logic in ***d2/d2-php/api/analyzer.php***

JS: Start the server by navigating to the folder and use ***node server.js***
JS: Place your server-side logic in ***d2/d2-js/api/analyzer.js***

Requirements

Create a server-side solution that:

- Accepts text input via POST request
- Validates that text is provided and non-empty
- Calculates and returns the following metrics:

Metric	Description
charCount	Total character count (including spaces)
charCountNoSpaces	Character count (excluding spaces)
wordCount	Total number of words

Response Format

Success Response:

```
{
  "success": true,
  "analysis": {
    "charCount": 500,
    "charCountNoSpaces": 420,
    "wordCount": 85
    ]
  }
}
```

Error Response

```
{
  "success": false,
  "message": "Text is required and must be non-empty"
}
```

Please refer to the sample folder and ***d2-sample.mp4 video***.

★ D3 - Contact Logger (L2)

Using JavaScript or PHP, implement a simple contact logger system that validates and stores contact messages in JSON format. You should not modify any of the existing code.

PHP: Place your server-side logic in ***d3/d3-php/api/contact.php***

JS: Start the server by navigating to the folder and use ***node server.js***
JS: Place your server-side logic in ***d3/d3-js/api/contact.js***

Requirements:

Create a server-side solution that:

- Accepts contact form data (name, email, message)
- Validates input according to specified rules
  - **Name**: minimum 2 characters
  - **Email**: valid email format
  - **Message**: minimum 10 characters
- Generate a random 8-digit ID
- Stores data persistently in JSON format (***data/contacts.json***)
- Returns appropriate JSON responses (below format), without overwriting the existing ones

```
{
  "id": "32226575",
  "name": "Val Adam",
  "email": "val@email.com",
  "message": "Hello world!"
},
```

Please refer to the sample folder and d3-sample.mp4 video.

★ D4 - Loading Animation (L1)

Using CSS only (no JavaScript allowed), create a loading spinner with text animation in the centre of the page.

Requirements:

- **Spinner**: 60px diameter circle with rotating border animation
  - Border: 8px solid, light grey (#f3f3f3) background with blue top border (#3498db)
  - Continuous rotation (2 second duration)
- **Loading text**: "Loading..." text that fades in and out
  - Font size: 18px, centred below spinner
  - Fade animation
- **Container**: Both elements centred on page vertically and horizontally

Please refer to the **sample folder** and ***d4_sample.mp4*** video.

★ D5 - Quiz Scorer (L1)

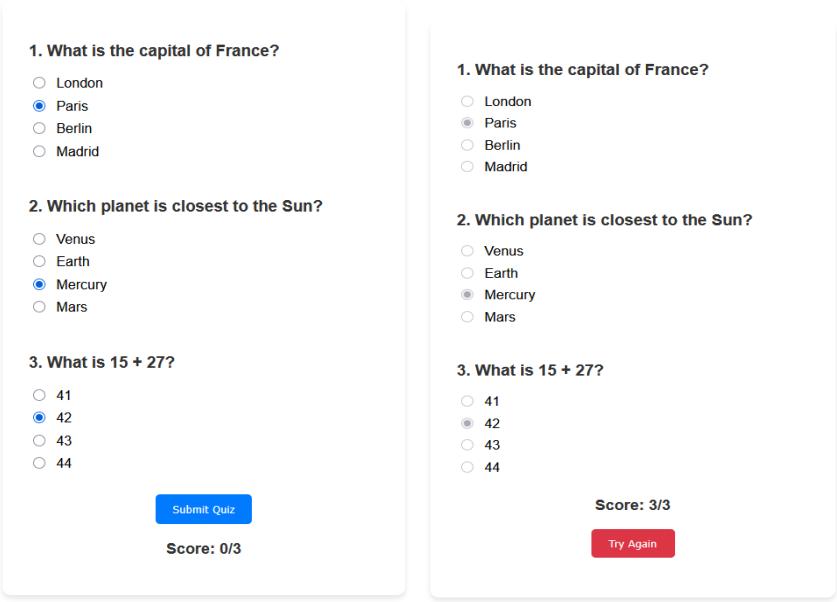
Using JavaScript only (do not modify the HTML/CSS), create a simple quiz scoring system. The quiz contains 3 multiple choice questions, and the score should update when the quiz is submitted.

Requirements

Create a quiz scoring system that:

- **Calculate Score**: When "Submit Quiz" button is clicked, calculate total score by summing values of selected radio buttons and display as "Score: X/3"
- **Disable Form**: After submission, disable all radio buttons and hide submit button, show "Try Again" button
- **Reset Functionality**: When "Try Again" is clicked, clear all selections, enable radio buttons, reset score to "Score: 0/3", and show submit button
- **Score Calculation**: Correct answers have value="1", incorrect answers have value="0" in

Place your code in ***`js/quiz.js`***



★ D6 - Neon Glow Button (L1)

Using CSS only (no JavaScript allowed), create an interactive neon-style button with animated glowing effects. The button should pulse with a neon glow and have additional hover effects.

The provided HTML should not be modified at all and the CSS do not modify above the designated area.

Place your CSS code in css/style.css

Requirements:

Create a neon glow button that:

- **Button Styling**: Button is ***200px wide***, ***60px*** tall, centred on page with transparent background, white text "*CLICK ME*", and 2px solid border (***#00ff88***) (button class is "***neon-button***")
- **Continuous Glow Animation**: Button has a pulsing neon glow using box-shadow that animates between 0px and 20px blur with green colour (***#00ff88***), animation duration 1.5s, infinite loop
- **Text Shadow Effect**: Text has matching animated text-shadow that pulses in sync with box-shadow
- **Hover State**: On hover, button background fills with semi-transparent green (#***00ff8844***), border becomes 3px, and glow intensifies to 30px blur
- **Smooth Transitions**: All hover effects use smooth transitions (0.3s duration)

Please refer to the sample folder and ***d6-sample.mp4*** video.

★ D7 – JS Functions (L2)

Using JavaScript only (no libraries), implement five small functions in ***js/functions.js*** file so that the index page shows PASS/FAIL for each test automatically. Do not modify any other of the provided files.

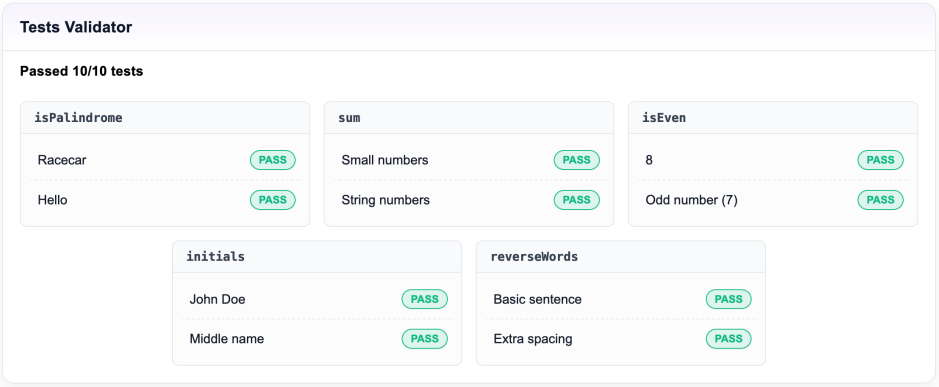
Requirements:

Implement the following 5 functions that will validate the functionality. Function names ***must match exactly*** as below (case-sensitive) in order to pass automatic tests.

1. ***isPalindrome(str)*** → ***Boolean***
   - Return true if the cleaned text (lower-case, alphanumeric only) reads the same forwards and backwards.
   - `"racecar"` → true, `"hello"` → false, `""` or non-string → false.
2. ***sum(a, b)*** → ***number***
   - Add two values and return a number. Accept numeric strings and negatives.
   - Examples: `(2,3)` → 5, `("5","10")` → 15, `(-5,10)` → 5.
3. ***isEven***(n) → ***Boolean***
   - Return true if n is an integer evenly divisible by 2.
   - `8` → true, `7` → false, `0` → true, `-4` → true, non-integer or NaN → false.
4. ***initials(fullName)*** → ***string***
   - Return uppercase initials with dots. Ignore extra spaces.
   - `"John Doe"` → "J.D.", `"Grace Murray Hopper"` → "G.M.H.", `"Cher"` → "C.".
5. ***reverseWords***(***sentence)*** → ***string***
   - Return the sentence with words in reverse order, trimming extra spaces.
   - `"one two three"` → "three two one", `" hello world "` → "world hello".

Notes

- Check the browser console for error messages if results do not appear.
- Each function is tested independently — one failing function will not affect the rest.



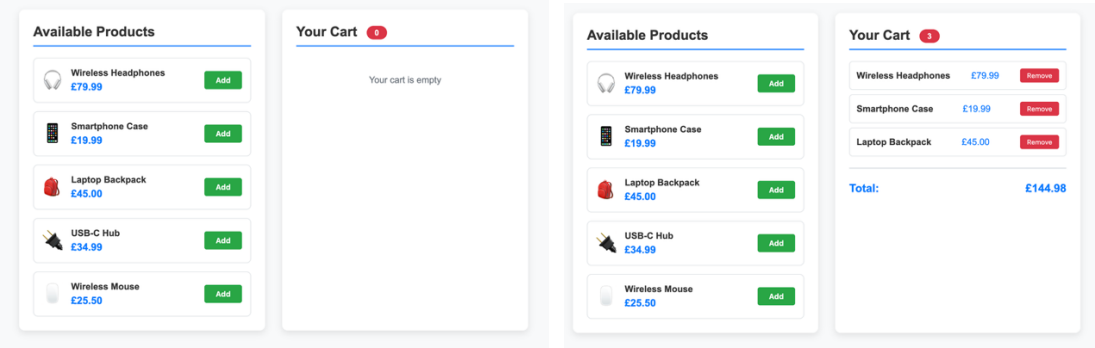
★ D8 – Shopping Cart (L3)

Using JavaScript only (no libraries), implement a shopping cart system that allows users to add items and view live price calculations.

JS: Place your code in ***`js/solution.js`***

Requirements

- Adds products to the cart when "Add to Cart" is clicked
- Allows removing items from the cart
- Updates the cart badge with total item count
- Displays cart items with name, and price
- Calculates and displays:
  - Total
- Shows "Your cart is empty" message when cart is empty



Please refer to the sample folder and ***`d8-sample.mp4 video`***.

★ D9 - Phone Number Formatter (L1)

Using JavaScript only (no libraries), implement a simple client-side phone number formatter that validates and formats UK phone numbers into different output styles using JavaScript in the file: js/formatter.js.

Requirements

Input Validation

- Only digits are allowed
- Validate UK phone rules:
  - Numbers starting with '0' must be exactly 11 digits
  - Numbers starting with '44' must be exactly 12 digits
  - Any other format is invalid

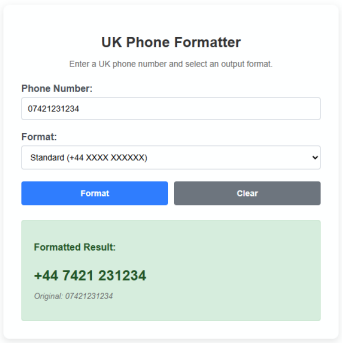
Output Formats

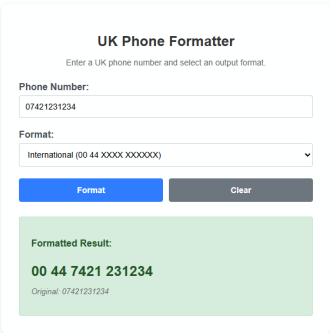
Format	Output Example
standard	*+44 7123 456789*
international	*00 44 7123 456789*

Return Format

```
Return Format
// Success
{ success: true, formatted: "+44 7123 456789"}

// Error
{ success: false, error: "Invalid phone number"}
```





★ D10 - Animated Logo Background (L1)

Using CSS only, create an animated gradient background effect behind the WorldSkills UK logo. The gradient should smoothly flow in multiple directions creating a dynamic visual effect.

Requirements

- Create a container that fills the main area and is centred
- Apply a random movement gradient using brand colours (***#003764*** navy, ***#DA291C*** red)
- Animate the gradient to flow smoothly in a continuous loop
- Position the logo image to cover the container

Please refer to the sample folder and ***d10-sample.mp4*** video.

General Guidance

Some files are asking specific to modify the code above or below. Please do not modify these files as might break the expected solution or invalidate it.

Task Submission

Place the ***index.html*** file that was provided in the ***dist/*** folder along with ***only completed tasks*** in the ***module-d repo***.

Marking Scheme Summary

SECTION	Level	TOTAL
D1	Level 1	2
D2	Level 3	3.75
D3	Level 2	2.75
D4	Level 1	2
D5	Level 1	2
D6	Level 1	2
D7	Level 2	2.75
D8	Level 3	3.75
D9	Level 1	2
D10	Level 1	2
TOTAL MARKS		**25**