CS4116 Week 2 Lab Guide

Software Development Project

Teaching Assistant Guide

Prepared by: Zia Ur Rehman
Course: CS4116 - Software Development Project
Professor: Conor Ryan

University of Limerick
2026

Contents

1 Introduction and Welcome

Welcome to Week 2 Lab! Today we'll be working through the fundamentals of HTML that will form the foundation for your web development project this semester. Before we dive into the technical content, let me remind you about some important course logistics.

2 Important Reminders: Groups and Project

2.1 Group Formation (Due: Friday of Week 2)

Option 1: Choose Your Own Group

- Form a group of **exactly 4 people**
- Send an email to **rehman.zia@ul.ie** by **Friday of Week 2**
- **Subject line format:** `[CS4116] Group <ID1> <ID2> <ID3> <ID4>`
- **CC all group members** in the email
- List all names in the email body

Option 2: Random Assignment

- If you don't form a group, you'll be entered into a random draw in Week 3
- **Bonus:** You get **2% extra credit** for taking your chances!
- You may submit an **Excuse List** to Conor.Ryan@ul.ie by Friday Week 2
  - Subject line: `[CS4116] Excuse List`
  - List up to 2 people you don't want to be grouped with
  - The teaching team will do their best to respect this

2.2 Project Overview

This is a **blended module** focused on building a web-based application. Key components:

- **HTML/CSS** - Structure and styling (starting now!)
- **Client/Server Architectures**
- **PHP** - Server-side programming
- **SQL** - Database management
- **Trello** - Project management

2.3 Weekly Workload Tracking

Starting in **Week 5**, your group will update a weekly spreadsheet showing:

- Work done by each person that week
- Breakdown of contributions
- Examples:

- All equal: 25/25/25/25
- Less balanced: 30/30/20/20
- No contribution: 33/33/33/0

Important!

This affects your individual grade! Your final project mark will be adjusted based on your average weekly contribution.

2.4 Key Deliverables to Remember

- **D2:** Website Pitch (Week 5) - 5%
- **D3:** Design Document (Week 6) - 10%
- **D4:** Database Implementation (Week 7) - 2%
- **Mid-term Exam (Week 9)** - 30% Includes coding questions!
- **D7:** Final Demo (Week 13) - 32%
- **Participation:** 5%

2.5 Important Note on AI Tools

As Professor Ryan emphasized:

> *"ChatGPT (or Gemini, or Claude, etc.) is not your lab partner!"*

- GenAI can produce plausible-looking code that is subtly wrong
- The mid-term will include coding questions **without AI access**
- Employers want coders who understand the fundamentals
- You will face coding tests in job interviews

How to succeed

Take notes, rewatch videos, read the Course Guide, and most importantly - **understand the code you write!**

3 Setting Up Your Development Environment

Before we start coding, you need the right tools. Let's set up a proper HTML development environment.

3.1 Choosing a Text Editor

For this course, I **strongly recommend Visual Studio Code (VS Code)**. Here's why:

Why VS Code?

- Free and open-source
- Lightweight but powerful

- Excellent HTML/CSS support
- Live preview capability
- Huge extension ecosystem
- Cross-platform (Windows, Mac, Linux)

Download: https://code.visualstudio.com/
Alternatives (if you prefer):

- Sublime Text
- Atom
- Notepad++ (Windows only)
- Brackets

3.2 Essential VS Code Extensions

Once you have VS Code installed, add these extensions:

3.2.1 1. Live Server (Essential!)

- **Publisher:** Ritwick Dey
- **What it does:** Launches a local development server with live reload
- **Why you need it:** See your HTML changes instantly in the browser
- **How to install:**
  1. Open VS Code
  2. Click Extensions icon (or press Ctrl+Shift+X)
  3. Search for "Live Server"
  4. Click Install

3.2.2 2. HTML CSS Support (Recommended)

- **Publisher:** ecmel
- **What it does:** IntelliSense for HTML class names
- **Why it helps:** Auto-completion for CSS classes

3.2.3 3. Auto Rename Tag (Helpful)

- **Publisher:** Jun Han
- **What it does:** Automatically renames paired HTML tags
- **Why it helps:** When you change `<h1>` to `<h2>`, the closing tag updates automatically

3.2.4 4. Prettier - Code Formatter (Optional but recommended)

- **Publisher:** Prettier
- **What it does:** Automatically formats your code
- **Why it helps:** Keeps your code clean and readable

3.3 Creating Your Project Folder

Let's organize your work properly:

1. **Create a folder structure:**

```
Week02/
 myfile.html
```

2. **Open the folder in VS Code:**
   - File → Open Folder
   - Select your Week02 folder
3. **Create your first HTML file:**
   - Right-click in the Explorer pane
   - New File → `myfile.html`

3.4 File Naming Conventions

Important rules

- Use `.html` extension (not `.htm`)
- No spaces in filenames (use `my_file.html` or `myfile.html`, not `my file.html`)
- Keep names lowercase for consistency
- Make names descriptive but concise

4 HTML Fundamentals Review

Before we tackle the lab tasks, let's review the essential HTML concepts you'll need.

4.1 What is HTML?

HTML = HyperText Markup Language

- **HyperText:** Text with links to other text
- **Markup:** Tags that define structure and meaning
- **Language:** A system of communication (between you and the browser)

Key points:

- HTML describes **content and structure**
- The browser **interprets and renders** HTML
- Different browsers use different rendering engines:
  - Chrome/Opera: Blink
  - Firefox: Gecko
  - Safari: WebKit

4.2 Elements vs. Tags

Understanding the terminology:

- **Tag:** An opening or closing entity
  - Opening tag: `<p>`
  - Closing tag: `</p>`
- **Element:** A complete structure with tags and content
  - `<p>This is a paragraph</p>` ← This is an element
- **Self-closing elements:** Some elements don't need closing tags
  - `<hr>` - Horizontal rule
  - `<br>` - Line break
  - `<img>` - Image

4.3 Basic HTML Document Structure

Every HTML document follows this template:

Listing 1: Basic HTML Template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale
        =1.0">
    <title>Page Title Here</title>
</head>
<body>
    <!-- Your visible content goes here -->
</body>
</html>
```

Let's break this down:

1. `<!DOCTYPE html>`
   - Tells the browser this is an HTML5 document
   - Must be the very first line
   - Not case-sensitive, but use lowercase by convention
2. `<html lang="en">`
   - Root element that contains all content
   - `lang="en"` specifies English language (helps screen readers and search engines)
3. `<head>` **section**
   - Contains **metadata** (information about the page)
   - Not visible to users
   - Includes title, character encoding, stylesheets, scripts

4. `<meta charset="UTF-8">`
   - Specifies character encoding
   - UTF-8 supports all languages and special characters
5. `<meta name="viewport">`
   - Important for responsive design
   - Ensures proper rendering on mobile devices
6. `<title>` **element**
   - Appears in browser tab
   - Used by search engines
   - Should be descriptive and unique
7. `<body>` **section**
   - Contains all **visible content**
   - Everything users see appears here

4.4 Comments in HTML

Comments help you document your code:

Listing 2: HTML Comments

```
<!-- This is a comment -->
<!-- Comments are ignored by the browser -->
<!--
    Multi-line comment:
    You can write notes here
    for yourself or other developers
-->
```

Best practices

- Use comments to explain complex sections
- Document why you did something (not just what)
- Remove comments before deploying to production (optional)

5 Task 1: Creating a Basic HTML Page

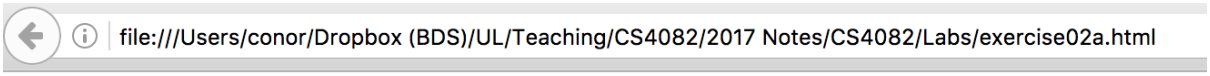

Figure 1: Reference output for Task 1

5.1 Understanding the Requirements

What you need to create:

- A file named `myfile.html`
- Should look like the provided image
- Must include:
  - Main heading
  - Subheading
  - Smallest possible heading
  - Paragraphs with text
  - A hyperlink

5.2 HTML Concepts You'll Need

5.2.1 Concept 1: Headings

HTML provides **six levels of headings**:

Listing 3: HTML Headings

```
<h1>This is a Heading 1 (Largest)</h1>
<h2>This is a Heading 2</h2>
<h3>This is a Heading 3</h3>
<h4>This is a Heading 4</h4>
<h5>This is a Heading 5</h5>
<h6>This is a Heading 6 (Smallest)</h6>
```

When to use each:

- `<h1>`: Main page title (use only once per page)
- `<h2>`: Major sections
- `<h3>`: Subsections
- `<h4>-<h6>`: Further subdivisions (less common)

Important

Headings are **semantic elements** - they convey meaning, not just appearance.

Example:

Listing 4: Heading Hierarchy

```
<h1>My Research Paper</h1>
<h2>Introduction</h2>
<h3>Background</h3>
<h3>Motivation</h3>
<h2>Methodology</h2>
<h3>Data Collection</h3>
```

5.2.2 Concept 2: Paragraphs

The `<p>` element defines a paragraph:

Listing 5: Paragraphs

```
<p>This is a paragraph of text.</p>
<p>This is another paragraph.</p>
```

Key points:

- Browsers automatically add space between paragraphs
- Text wraps automatically to fit the container
- Multiple spaces and line breaks in your code are collapsed to a single space

Example of space collapsing:

Listing 6: Space Collapsing

```
<!-- This: -->
<p>This    has    many    spaces</p>

<!-- Displays as: -->
This has many spaces
```

5.2.3 Concept 3: Hyperlinks

The `<a>` (anchor) element creates links:

Listing 7: Basic Link

```
<a href="http://google.com">Click here</a>
```

Anatomy:

- `<a>` - Opening tag
- `href="URL"` - Attribute specifying the destination
- `Click here` - The clickable text (link text)
- `</a>` - Closing tag

Types of links:

1. **External links** (to other websites):

```
<a href="https://www.google.com">Google</a>
<a href="http://www.ul.ie">University of Limerick</a>
```

2. **Internal links** (to other pages in your site):

```
<a href="about.html">About Us</a>
<a href="contact.html">Contact</a>
```

3. **Anchor links** (to sections on the same page):

```
<a href="#section1">Jump to Section 1</a>
<!-- Later on the page: -->
<h2 id="section1">Section 1</h2>
```

4. **Email links:**

```
<a href="mailto:someone@example.com">Email me</a>
```

Best practices for link text

- Bad: "Click here"
- Good: "Download the course syllabus"
- Make link text descriptive and meaningful

5.2.4 Concept 4: Horizontal Rule

The `<hr>` element creates a horizontal line:

Listing 8: Horizontal Rule

```
<hr>
```

Use cases:

- Separate sections of content
- Visual break in the page
- Thematic transition

Note

This is a **self-closing** element (no closing tag needed).

5.3 Complete Solution with Comments

Listing 9: Task 1 Complete Solution

```
<!DOCTYPE html>
<!-- Declares this as an HTML5 document -->

<html lang="en">
<!-- Root element; lang attribute specifies English -->

<head>
    <!-- Head section contains metadata not visible to users -->

    <meta charset="UTF-8">
    <!-- Sets character encoding to UTF-8 for international character
        support -->

    <meta name="viewport" content="width=device-width, initial-scale
        =1.0">
    <!-- Ensures responsive design on mobile devices -->

    <title>This is the page title</title>
    <!-- Title appears in browser tab and search results -->
</head>

<body>
    <!-- Body section contains all visible content -->

    <h1>This is the page title</h1>
    <!-- h1 is the main heading - use only once per page -->

    <h2>Here's a smaller title</h2>
    <!-- h2 is used for major sections -->

    <h6>This is the smallest possible title</h6>
    <!-- h6 is the smallest heading level available -->

    <p>Paragraphs should be contained within their own tags, and can
        contain <a href="http://google.com">links</a> in the middle.</p>
    <!-- p element defines a paragraph; a element creates a hyperlink
        with href attribute specifying destination -->

    <p>There's always space between paragraphs, to make the page easier
         to read</p>
    <!-- Browsers automatically add vertical spacing between paragraph
        elements -->
</body>

</html>
<!-- Closing html tag marks the end of the document -->
```

5.4 Testing Your Page

Using Live Server (Recommended):

1. Right-click on `myfile.html` in VS Code
2. Select "Open with Live Server"

3. Your default browser will open with your page
4. Any changes you save will automatically refresh the browser!

Manual Method:

1. Save `myfile.html`
2. Double-click the file in your file explorer
3. It will open in your default browser
4. Refresh (F5) after making changes

5.5 Common Mistakes to Avoid

Mistake 1: Forgetting closing tags

```
<!-- Wrong -->
<p>This is a paragraph
    <p>This is another paragraph

<!-- Correct -->
<p>This is a paragraph</p>
   <p>This is another paragraph</p>
```

Mistake 2: Nesting tags incorrectly

```
<!-- Wrong -->
<h1><p>Wrong nesting</h1></p>

<!-- Correct -->
<h1>Correct heading</h1>
   <p>Correct paragraph</p>
```

Mistake 3: Forgetting quotes in attributes

```
<!-- Wrong -->
<a href=http://google.com>Link</a>

<!-- Correct -->
<a href="http://google.com">Link</a>
```

Mistake 4: Using wrong heading hierarchy

```
<!-- Wrong -->
<h1>Main Title</h1>
   <h4>Should be h2</h4>

<!-- Correct -->
<h1>Main Title</h1>
   <h2>Proper subsection</h2>
```

6 Task 3: Creating HTML Tables

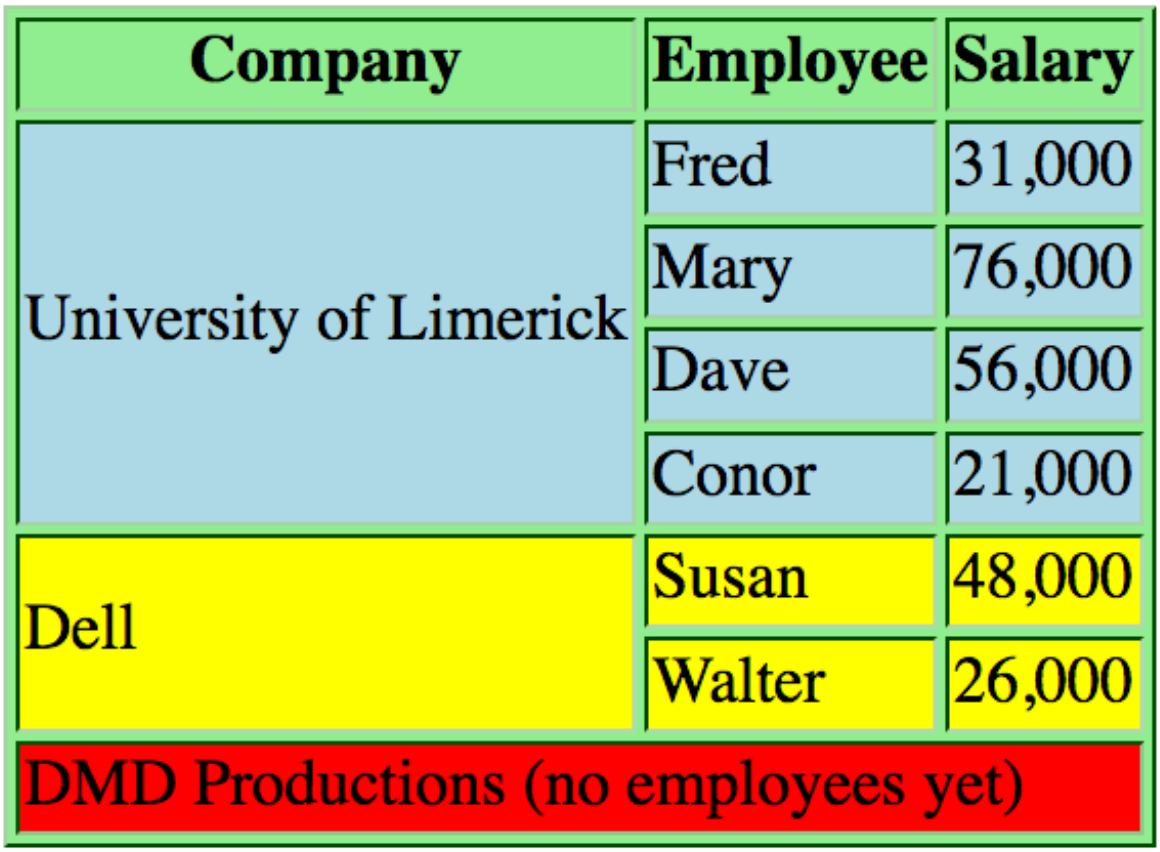

Figure 2: Reference output for Task 3

6.1 Understanding the Requirements

You need to create a table that matches the provided image:

- Company column (with merged cells)
- Employee and Salary columns
- Different background colors for different companies
- Specific data for University of Limerick, Dell, and DMD Productions

6.2 HTML Concepts You'll Need

6.2.1 Concept 1: Basic Table Structure

HTML tables use a hierarchical structure:

Listing 10: Basic Table

```
<table>
    <tr>
        <td>Cell 1</td>
        <td>Cell 2</td>
    </tr>
```

```
    <tr>
        <td>Cell 3</td>
        <td>Cell 4</td>
    </tr>
</table>
```

Key elements:

- `<table>` - Container for the entire table
- `<tr>` - Table Row
- `<td>` - Table Data (a cell)
- `<th>` - Table Header (bold and centered by default)

Important

HTML tables are **row-based**. You define rows, and put cells in each row. There's no explicit "column" element!

Simple example:

Listing 11: Table with Headers

```
<table border="1">
    <tr>
        <th>Name</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Alice</td>
        <td>25</td>
    </tr>
    <tr>
        <td>Bob</td>
        <td>30</td>
    </tr>
</table>
```

6.2.2 Concept 2: Table Attributes

HTML tables support various attributes to control appearance:

1. Border:

```
<table border="1">
```

- Sets border width in pixels
- `border="0"` means no border
- Modern practice: Use CSS instead, but border attribute is quick for learning

2. Cell Padding:

```
<table cellpadding="5">
```

- Space between cell content and cell border

- Makes content easier to read
- Value in pixels

3. Cell Spacing:

```
<table cellspacing="5">
```

- Space between cells
- Creates gaps between table cells
- Value in pixels

4. Background Color:

```
<table bgcolor="green">
```

- Sets background color for entire table
- Can use color names or hex codes (#00FF00)

Can also be applied to individual rows or cells:

Listing 12: Row and Cell Colors

```
<tr bgcolor="red">
    <td bgcolor="blue">Blue cell</td>
    <td>Red cell (from row)</td>
</tr>
```

6.2.3 Concept 3: Spanning Rows and Columns

Column Span (colspan):
Makes a cell stretch across multiple columns

Listing 13: Column Span Example

```
<table border="1">
    <tr>
        <td colspan="2">This cell spans 2 columns</td>
    </tr>
    <tr>
        <td>Normal cell 1</td>
        <td>Normal cell 2</td>
    </tr>
</table>
```

Row Span (rowspan):
Makes a cell stretch across multiple rows

Listing 14: Row Span Example

```
<table border="1">
    <tr>
        <td rowspan="2">Spans 2 rows</td>
        <td>Row 1, Cell 2</td>
    </tr>
    <tr>
        <td>Row 2, Cell 2</td>
    </tr>
</table>
```

Key insight for rowspan

When a cell spans multiple rows, you **don't include a cell in that position** in subsequent rows!

6.3 Planning the Table

Before coding, let's analyze the required table:

Structure:

```
+----------------------+----------+--------+
| Company              | Employee | Salary |
+----------------------+----------+--------+
| University of        | Fred     | 31,000 |
| Limerick             | Mary     | 76,000 |
| (spans 4 rows)       | Dave     | 56,000 |
|                      | Conor    | 21,000 |
+----------------------+----------+--------+
| Dell                 | Susan    | 48,000 |
| (spans 2 rows)       | Walter   | 26,000 |
+----------------------+----------+--------+
| DMD Productions (no employees yet)      |
+----------------------+----------+--------+
```

Color scheme:

- University of Limerick: Light blue/cyan background
- Dell: Yellow background
- DMD Productions: Red background
- Header row: Green background

6.4 Complete Solution with Comments

Listing 15: Task 3 Complete Solution

```
<!DOCTYPE html>
<!-- HTML5 document declaration -->

<html lang="en">
<!-- Root element with English language specification -->

<head>
    <meta charset="UTF-8">
    <!-- Character encoding for proper text display -->

    <meta name="viewport" content="width=device-width, initial-scale
        =1.0">
    <!-- Responsive design viewport settings -->

    <title>Company Employee Table</title>
    <!-- Page title shown in browser tab -->
</head>

```

```
<body>
    <table border="1">
    <!-- Table element with 1-pixel border -->

        <tr bgcolor="lightgreen">
        <!-- Table row with light green background color -->
            <th>Company</th>
            <!-- Table header cell for Company column -->
            <th>Employee</th>
            <!-- Table header cell for Employee column -->
            <th>Salary</th>
            <!-- Table header cell for Salary column -->
        </tr>

        <!-- University of Limerick Section -->
        <tr bgcolor="lightblue">
        <!-- First row for UL with light blue background -->
            <td rowspan="4" bgcolor="lightblue">University of Limerick</td>
            <!-- Company cell spans 4 rows (for 4 employees); light blue background -->
            <td>Fred</td>
            <!-- Employee name cell -->
            <td>31,000</td>
            <!-- Salary cell -->
        </tr>

        <tr bgcolor="lightblue">
        <!-- Second UL employee row - only 2 cells because Company cell spans from above -->
            <td>Mary</td>
            <!-- Employee name -->
            <td>76,000</td>
            <!-- Salary -->
        </tr>

        <tr bgcolor="lightblue">
        <!-- Third UL employee row -->
            <td>Dave</td>
            <td>56,000</td>
        </tr>

        <tr bgcolor="lightblue">
        <!-- Fourth UL employee row -->
            <td>Conor</td>
            <td>21,000</td>
        </tr>

        <!-- Dell Section -->
        <tr bgcolor="yellow">
        <!-- First Dell row with yellow background -->
            <td rowspan="2" bgcolor="yellow">Dell</td>
            <!-- Company cell spans 2 rows (for 2 employees) -->
            <td>Susan</td>
            <td>48,000</td>
        </tr>

        <tr bgcolor="yellow">
```

```
        <!-- Second Dell employee row - only 2 cells -->
            <td>Walter</td>
            <td>26,000</td>
        </tr>

        <!-- DMD Productions Section -->
        <tr>
            <td colspan="3" bgcolor="red">DMD Productions (no employees
                yet)</td>
            <!-- Single cell spanning all 3 columns; red background
                indicates no employees -->
        </tr>

    </table>
</body>

</html>
```

6.5 Understanding Rowspan Logic

This is the trickiest part!

Let's visualize how rowspan works:

Without rowspan:

```
Row 1: [Cell 1] [Cell 2] [Cell 3]
Row 2: [Cell 1] [Cell 2] [Cell 3]
Row 3: [Cell 1] [Cell 2] [Cell 3]
```

With rowspan="3" on Row 1, Cell 1:

```
Row 1: [Cell 1 - spans down] [Cell 2] [Cell 3]
Row 2: [   still Cell 1    ] [Cell 2] [Cell 3]
Row 3: [   still Cell 1    ] [Cell 2] [Cell 3]
```

In code:

Listing 16: Rowspan Logic

```
<tr>
    <td rowspan="3">Cell 1</td>  <!-- This cell spans 3 rows -->
    <td>Cell 2</td>
    <td>Cell 3</td>
</tr>
<tr>
    <!-- NO first cell here! It's still being used from row 1 -->
    <td>Cell 2</td>
    <td>Cell 3</td>
</tr>
<tr>
    <!-- NO first cell here either! -->
    <td>Cell 2</td>
    <td>Cell 3</td>
</tr>
```

Common mistake

```
<!-- WRONG: -->
<tr>
    <td rowspan="2">Spans 2 rows</td>
    <td>Cell 2</td>
</tr>
<tr>
    <td>This cell will be EXTRA!</td>  <!-- ERROR: The first cell
        is still spanning! -->
    <td>Cell 2</td>
</tr>

<!-- CORRECT: -->
<tr>
    <td rowspan="2">Spans 2 rows</td>
    <td>Cell 2</td>
</tr>
<tr>
    <!-- First cell is still active from above -->
    <td>Cell 2</td>
</tr>
```

6.6 Testing and Refinement

Test your table:

1. Open in browser using Live Server
2. Check:
    - All borders visible?
    - Colors correct?
    - Data accurate?
    - Cells spanning correctly?

Common issues:

1. **Extra cell appears**
    - **Cause:** You included a cell in a row where another cell is still spanning
    - **Fix:** Remove the extra `<td>` element
2. **Missing cell (gap in table)**
    - **Cause:** You forgot to include a cell, or rowspan value is wrong
    - **Fix:** Check your rowspan numbers and ensure all cells are accounted for
3. **Colors not showing**
    - **Cause:** Typo in color name or missing quotes
    - **Fix:** Use standard color names (red, blue, yellow, lightblue, lightgreen) with quotes

7 Additional Tips and Best Practices

7.1 HTML Validation

Your HTML should be **valid** according to web standards:

Common validation errors:

1. Unclosed tags
2. Tags in wrong order
3. Missing required attributes
4. Incorrect nesting

How to validate:

- Use W3C Markup Validation Service: https://validator.w3.org/
- Or use VS Code extensions like "HTMLHint"

7.2 Code Readability

Use proper indentation:

Listing 17: Readable vs Unreadable Code

```
<!-- Hard to read: -->
<table><tr><td>Cell 1</td><td>Cell 2</td></tr><tr><td>Cell 3</td><td>
    Cell 4</td></tr></table>

<!-- Easy to read: -->
<table>
    <tr>
        <td>Cell 1</td>
        <td>Cell 2</td>
    </tr>
    <tr>
        <td>Cell 3</td>
        <td>Cell 4</td>
    </tr>
</table>
```

Indentation rules

- Indent child elements one level
- Use consistent spacing (2 or 4 spaces, or tabs)
- VS Code can auto-format (Right-click → Format Document)

7.3 Semantic HTML

Use meaningful elements:

Listing 18: Semantic vs Non-Semantic

```
<!-- Less semantic: -->
<div class="header">Page Title</div>
```

```
3
4 <!-- More semantic: -->
5 <h1>Page Title</h1>
```

Why it matters:

- Screen readers understand your content better
- Search engines can better index your page
- Code is more maintainable
- Accessibility is improved

7.4 Common HTML Entities

Sometimes you need to display special characters:

Entity	Character	Description
`<`	¡	Less than
`>`	¿	Greater than
`&`	&	Ampersand
`&quot;`	”	Quote
` `	(space)	Non-breaking space
`&copy;`	©	Copyright

Table 1: Common HTML Entities

Example:

Listing 19: Using HTML Entities

```
1 <p>The tag &lt;p&gt; creates a paragraph.</p>
2 <!-- Displays: The tag <p> creates a paragraph. -->
```

8 Troubleshooting Guide

8.1 Problem 1: “My page shows code instead of rendering”

Cause: File might be saved as .txt instead of .html **Solution:** Ensure filename ends with `.html`

8.2 Problem 2: “Changes don’t appear when I refresh”

Cause: Browser cache or file not saved **Solution:**

- Make sure to save (Ctrl+S)
- Hard refresh (Ctrl+F5)
- Use Live Server for automatic updates

8.3 Problem 3: "Link doesn't work"

Cause: Missing `http://` or `https://` in href **Solution:**

```
<!-- Wrong -->
<a href="google.com">Link</a>

<!-- Correct -->
<a href="http://google.com">Link</a>
```

8.4 Problem 4: "Table cells are misaligned"

Cause: Incorrect rowspan/colspan or missing cells **Solution:**

- Count cells in each row
- Verify rowspan/colspan values
- Check that rows with spanning cells have fewer explicit cells

8.5 Problem 5: "Colors not showing"

Cause: Typo in color name or attribute **Solution:**

```
<!-- Wrong -->
<td backcolor="red">
<td bgcolor=red>

<!-- Correct -->
<td bgcolor="red">
```

9 Going Beyond: Optional Enhancements

If you finish early, try these challenges:

9.1 Challenge 1: Add More Styling

Listing 20: Enhanced Table Styling

```
<table border="1" cellpadding="10" cellspacing="0">
    <!-- cellpadding adds space inside cells -->
    <!-- cellspacing=0 removes gaps between cells -->
</table>
```

9.2 Challenge 2: Add Table Caption

Listing 21: Table with Caption

```
<table border="1">
    <caption>Company Employee Directory</caption>
    <!-- Caption appears above the table -->
    <tr>
        <th>Company</th>
        <th>Employee</th>
        <th>Salary</th>
    </tr>
```

```
    <!-- rest of table -->
</table>
```

9.3 Challenge 3: Experiment with More Links

Listing 22: Advanced Link Usage

```
<p>
    Visit our <a href="http://www.ul.ie" target="_blank">university
        website</a>.
    <!-- target="_blank" opens link in new tab -->
</p>

<p>
    Email us at <a href="mailto:info@ul.ie">info@ul.ie</a>
</p>
```

9.4 Challenge 4: Add More Semantic Elements

Listing 23: Semantic HTML5 Elements

```
<header>
    <h1>My Page</h1>
    <nav>
        <a href="home.html">Home</a> |
        <a href="about.html">About</a>
    </nav>
</header>

<main>
    <article>
        <h2>Article Title</h2>
        <p>Content here...</p>
    </article>
</main>

<footer>
    <p>&copy; 2026 My Website</p>
</footer>
```

10 Summary and Next Steps

10.1 What We Covered Today

Groups and project logistics

- Group formation (due Friday Week 2)
- Weekly workload tracking
- Key deliverables and timeline

Development environment setup

- Installing VS Code

- Essential extensions (Live Server)
- Project organization

HTML fundamentals

- Document structure
- Headings and paragraphs
- Links and attributes
- Tables with rowspan and colspan

Two complete lab solutions

- Task 1: Basic HTML page with headings, paragraphs, and links
- Task 3: Complex table with merged cells and colors

10.2 Key Takeaways

1. **HTML is about structure and meaning**, not appearance
2. **Every element has opening and closing tags** (except self-closing ones)
3. **Attributes modify element behavior** (href, bgcolor, rowspan, etc.)
4. **Rowspan and colspan require careful planning** - remember to skip cells!
5. **Proper indentation makes code readable** and maintainable

10.3 What's Next

This week:

- Complete the lab tasks
- Form your group (if choosing your own)
- Review the video segments on Brightspace
- Start thinking about project ideas

Week 3:

- Tutorials begin (Q&A sessions)
- New lab sheet released
- Groups finalized
- More advanced HTML and introduction to CSS

10.4 Resources

Course materials:

- Brightspace: https://learn.ul.ie
- Course Guide: Contains all deliverables and rubrics
- HTML Cheat Sheet: Available on Brightspace

External resources:

- W3Schools HTML Tutorial: https://www.w3schools.com/html/
- MDN Web Docs: https://developer.mozilla.org/en-US/docs/Web/HTML
- HTML Validator: https://validator.w3.org/

Getting help:

- Lab sessions: Ask TAs (Zia Rehman, Mritunjay Musale)
- Tutorials (Week 3 onwards): Q&A sessions
- Email: rehman.zia@ul.ie (include [CS4116] in subject)
- Professor Conor: Conor.Ryan@ul.ie

11 Final Reminders

11.1 Action Items for This Week

Complete Task 1: Create myfile.html

Complete Task 3: Create the employee table

Test both files in your browser

Validate your HTML (optional but recommended)

Send group formation email by Friday (if forming your own group)

Review lecture videos on Brightspace

Read through the Course Guide

11.2 Important Dates

- **Friday, Week 2:** Group formation deadline
- **Week 3:** Groups posted, Tutorials begin
- **Week 5:** D2 - Website Pitch (5%)
- **Week 9:** Mid-term exam with coding questions (30%)

11.3 Remember

> *"Take notes! Do not rely solely on the videos/PDFs. Rewatch the video segments. Read the Course Guide. There should be no surprises. Respect your group! There is nowhere to hide in the group meetings..."*
> — Professor Conor Ryan

The mid-term will include coding questions. The best way to prepare is to **practice, practice, practice!** Don't rely on AI tools - understand the fundamentals.

Good luck with your lab work!

If you have any questions, don't hesitate to ask during the lab session or reach out via email.

Happy coding!

Prepared by Zia Ur Rehman - CS4116 Teaching Assistant
University of Limerick - 2026