

Unit 2: User Interface Design with HTML, CSS, and Bootstrap

Unit 2 — Moodle Introduction

Unit 2 moves from conceptual understanding to hands-on creation. You will learn to design professional user interfaces by applying HTML5 structure, CSS3 styling, and the Bootstrap framework. By the end of this unit, you will have built a fully styled, responsive login page and dashboard prototype for an eLearning system. All practicals must be submitted through the Moodle Workshop activity so that peers can provide structured feedback.

MULTIMEDIA

Animation Principle (Mayer, 2009): Unit 2 includes a short animation (created in Adobe Animate or CSS transitions) demonstrating how the Bootstrap grid reflows from three columns to one column as the browser window narrows. Animation is used here because the concept is inherently dynamic and cannot be conveyed as effectively by static text or images alone. The animation is embedded in the H5P Interactive Video and available as a standalone GIF for offline use.

Unit Aim

The aim of Unit 2 is to develop students' ability to design and implement professional, accessible, and responsive user interfaces for web-based eLearning applications, progressing from wireframe design through HTML structure to CSS styling and Bootstrap-based layout.

Intended Learning Outcomes

- Create wireframe designs for web-based eLearning interfaces using industry-standard design tools including Figma, Adobe XD, or Canva (Bloom's Level 3 — Create).
- Construct semantically correct, accessible HTML5 documents implementing modern structural elements and form controls (Bloom's Level 3 — Create).
- Apply CSS3 properties and the Bootstrap 5 grid system to produce responsive, visually professional web interfaces (Bloom's Level 3 — Apply).
- Evaluate a web interface design against WCAG 2.1 Level AA accessibility criteria and document required improvements (Bloom's Level 5 — Evaluate).

UDL NOTE

Multiple Means of Representation (UDL Guideline 2.5): All HTML and CSS code examples in this unit are presented in three ways—written in these notes, demonstrated live in the H5P video, and provided as downloadable starter files in Moodle. This reduces barriers for learners who prefer reading, watching, or hands-on tinkering as their primary mode of learning new syntax.

2.1 UI/UX Design Principles for eLearning Interfaces

Principle	Definition and Application
Clarity	Every element should have a single, clear purpose. Avoid visual clutter. Students should never need to wonder what a button does.

Consistency	Use the same colours, typography, and interaction patterns throughout the application. Consistency reduces cognitive load.
Feedback	The interface must respond to every user action. A button click should produce a visible change; a form submission should confirm success or explain errors.
Accessibility	Design for all users from the start, including those with visual, auditory, motor, or cognitive disabilities. Follow WCAG 2.1 Level AA.
Hierarchy	Use visual hierarchy (size, weight, colour, spacing) to guide the learner's eye to the most important information first.
Simplicity	Remove every element that does not contribute to the learner's goal. 'Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away' (Saint-Exupéry).

2.2 Wireframing with Figma, Adobe XD, and Canva

A wireframe is a low-fidelity visual blueprint of a web interface that shows the layout of elements without specifying final colours, typography, or imagery. Wireframing before coding is an essential professional practice: it forces design decisions to be made inexpensively, before any code is written, and creates a shared reference for team discussion and client approval.

2.2.1 Wireframing Tools

- Figma: The industry standard for collaborative UI/UX design. Browser-based, real-time collaboration, extensive component libraries. Free for individual use. <https://www.figma.com>
- Adobe XD: Professional vector-based design tool with prototyping features. Available as part of Adobe Creative Cloud.
- Canva: Accessible, template-based design tool suitable for educational contexts. Free tier available. <https://www.canva.com>
- InVision Studio: Prototyping-focused tool with animation capabilities. Used for creating interactive mockups that simulate real application behaviour.

INTERACTIVE

Adobe Captivate / Articulate Storyline Activity: Practical 2A requires students to submit a wireframe. Before submission, they interact with a Captivate-authored self-check activity that presents four wireframe screenshots and asks them to identify which violates a specific WCAG criterion. This reinforces the link between wireframing decisions and accessibility compliance before costly HTML coding begins.

2.2.2 Wireframe Components for an eLearning Login Screen

- Institution logo and name (top, centred): Establishes identity and trust. Accessibility requirement: the logo image must have an alt attribute.
- Username/email input field with visible label: Do NOT use placeholder text as a label—placeholders disappear on input and fail WCAG 2.1 Success Criterion 1.3.5.
- Password input field with show/hide toggle: Reduces input errors while maintaining security. The toggle must be keyboard-operable.

- 'Remember me' checkbox with label: Optional convenience feature; must not be pre-checked by default for security.
- Primary login button: Clear, high-contrast, minimum touch target size of 44×44px (WCAG 2.5.5).
- Forgot password link: Secondary action, visually distinct from the primary button.
- Error message region: Must be announced to screen readers via ARIA live region.

2.3 HTML5: Structure and Semantics

HTML (HyperText Markup Language) is the standard markup language for creating the structure of web pages. The current version, HTML5, introduced a set of semantic elements that describe the meaning of content—not just its appearance. Using semantic HTML is essential for accessibility: screen readers and assistive technologies rely on semantic structure to present content to users who cannot see the visual layout.

2.3.1 The Anatomy of an HTML Document

- `<!DOCTYPE html>` — Declares the document as HTML5.
- `<html lang="rw">` — Root element. The lang attribute sets the document language (important for screen readers and translation).
- `<head>` — Contains metadata: character encoding, viewport settings, page title, and links to CSS files.
- `<body>` — Contains all visible content. Structured using semantic elements described below.

2.3.2 Semantic HTML5 Elements

Element	Purpose and Educational Context
<code><header></code>	The introductory section of the page or a section. Typically contains the site logo, navigation, and page title.
<code><nav></code>	A block of navigation links. Screen readers announce this region so users can skip to it.
<code><main></code>	The primary content area. There should be exactly one <code><main></code> element per page.
<code><section></code>	A thematically related group of content with its own heading. Used to divide the main content into logical areas.
<code><article></code>	Self-contained content that makes sense on its own (e.g., a forum post, a course card).
<code><aside></code>	Content tangentially related to the main content (e.g., a sidebar with related resources).
<code><footer></code>	The concluding section of the page or a section. Typically contains copyright, contact, and secondary navigation links.
<code><form></code>	A data entry container. All form controls must be inside a form element with appropriate action and method attributes.

<label>	Associates a text description with a form control. Every input must have an associated label element.
---------	---

MULTIMEDIA

Pedagogically Meaningful Graphic: An annotated HTML document screenshot (included in the Moodle unit resources as a high-resolution PNG with alt text) colour-codes each semantic element—<header> in blue, <main> in green, <footer> in grey—using the Signalling Principle (Mayer, 2009) to visually distinguish structural regions without requiring students to read every line of code first.

2.4 CSS3: Styling Web Interfaces

Cascading Style Sheets (CSS) is the language used to describe the presentation of HTML content. CSS3 is the current standard, providing rich visual styling capabilities including custom properties (variables), flexbox and grid layout, animations, and media queries for responsive design.

2.4.1 The CSS Box Model

- Content: The actual content (text, image) of the element.
- Padding: Space between the content and the border. Padding is inside the element and inherits the element’s background colour.
- Border: The visible edge of the element. Defined by border-width, border-style, and border-color.
- Margin: Space outside the border between this element and adjacent elements. Margins are transparent.

box-sizing: border-box

By default, CSS adds padding and border to an element’s declared width, which causes layout miscalculations. Always set box-sizing: border-box in your CSS reset so that width includes padding and border. Add: *, *::before, *::after { box-sizing: border-box; } at the top of every stylesheet.

2.4.2 CSS Specificity

Selector Type	Specificity and Best Practice
Inline style (style attribute)	Specificity: 1,0,0,0 — Highest priority. Avoid using inline styles; they are impossible to override without !important.
ID selector (#id)	Specificity: 0,1,0,0 — High priority. Use sparingly; prefer class selectors for reusable components.
Class selector (.class), attribute selector, pseudo-class	Specificity: 0,0,1,0 — Standard priority. Preferred for component styling.
Element selector (div, p, h1)	Specificity: 0,0,0,1 — Lowest priority. Use for base/reset styles only.

2.4.3 Responsive Design with Media Queries

A responsive web design adapts its layout to the screen size of the device displaying it. This is essential for eLearning applications, where students may access content on desktop computers, tablets, or smartphones. Media queries apply CSS rules conditionally based on screen width.

2.5 Bootstrap 5: Responsive Framework

Bootstrap 5 is the world’s most widely used open-source CSS framework. It provides a pre-built collection of responsive layout classes, form controls, navigation components, and UI utilities that significantly accelerate the development of professional web interfaces. For eLearning applications, Bootstrap’s accessibility-conscious components are particularly valuable.

Class Prefix	Breakpoint
col- (xs)	< 576px — Extra small (mobile phones)
col-sm-	≥ 576px — Small (large phones, small tablets)
col-md-	≥ 768px — Medium (tablets)
col-lg-	≥ 992px — Large (small desktops)
col-xl-	≥ 1200px — Extra large (large desktops)
col-xxl-	≥ 1400px — Extra extra large (wide screens)

Unit 2 Assessments — ABC Learning Design Framework

Activity	ABC Type	Description	Bloom’s Level	Weight
Wireframe Design Submission — E-tivity 1	Production / Investigation	Using Figma, Adobe XD, or Canva, design annotated wireframes for the login screen and student dashboard of an eLearning system. Each wireframe must include component labels, rationale notes, and an accessibility checklist.	Create (L6)	10% (ICA)
HTML/CSS Practical — E-tivity 2	Practice / Production	Implement the wireframes from Activity 2A using HTML5 and CSS3. The submission must include semantic markup, visible focus styles, ARIA labels on all form inputs, and a linked external CSS file. Submit via Moodle Workshop.	Apply (L3)	10% (ICA)
Bootstrap Responsive Layout — E-tivity 3	Practice / Production	Extend your HTML/CSS submission by integrating Bootstrap 5. The layout must collapse to a single-	Apply (L3)	5% (ICA)

		column view on screens below 768px. Submit HTML and screenshot evidence at three breakpoints.		
WCAG 2.1 Audit Report	Investigation / Discussion	Using the WAVE browser extension, audit a peer's submitted HTML interface. Write a 400-word report identifying at least three accessibility issues and recommending specific code-level corrections.	Evaluate (L5)	5% (ICA)
Unit 2 Quiz — E-assessment 1	Practice	A 10-question Moodle Quiz covering HTML5 semantic elements, CSS specificity, the Bootstrap grid, and WCAG 2.1 criteria. Two attempts allowed; higher score recorded.	Remember / Apply	5% (ICA)

ASSESSMENT

Peer Assessment via Moodle Workshop: The Workshop activity is used for both the HTML/CSS Practical and the Bootstrap submission. Peer assessment develops critical evaluation skills (Bloom's Level 5—Evaluate) while simultaneously reducing the marking burden on the instructor. The Workshop rubric is identical to the instructor rubric (provided below), ensuring transparency and fairness.

End-Unit Assessment 1: Moodle Quiz — UI Design, HTML5, CSS3 & Bootstrap

□ Ready-to-Use Moodle Quiz — Copy & Paste Questions

10 questions. Set: Attempts = 2, Grading = Highest, Shuffle = Yes, Feedback = Immediate. Enter via Quiz > Edit Quiz > Add a new question.

Question & Answer	Notes
Q1. Which HTML5 element should contain the primary navigation links of an eLearning application? ✓ A) <nav> B) <section> C) <div> D) <aside> Feedback: The <nav> element is the semantic container for navigation links. Screen readers can jump directly to the <nav> region, improving keyboard accessibility.	
Q2. A student's form input has id="student-email". Which HTML attribute on the <label> element correctly associates it with this input? ✓ A) for="student-email" B) id="student-email" C) name="student-email" D) aria-label="student-email" Feedback: The for attribute on <label> must match the id of the associated input. This association is required for screen reader accessibility and click-to-focus behaviour.	

Q3. You have written <code>.card { color: red; }</code> and <code>#hero { color: blue; }</code>. An element with <code>class='card'</code> and <code>id='hero'</code> will display in which colour? ✓ A) Blue B) Red C) Neither D) It depends on the browser Feedback: ID selectors (0,1,0,0) have higher specificity than class selectors (0,0,1,0), so <code>#hero { color: blue; }</code> wins.	
Q4. Which Bootstrap 5 class combination makes a column occupy the full width on mobile and half the width on medium screens and above? ✓ A) <code>col-12 col-md-6</code> B) <code>col-6 col-md-12</code> C) <code>col-md-6</code> D) <code>col-sm-12 col-md-6</code> Feedback: Mobile-first Bootstrap: <code>col-12</code> sets full width as the default. <code>col-md-6</code> overrides this to half-width (6/12 columns) at the medium breakpoint (≥768px) and above.	
Q5. Why must the <code>lang</code> attribute be set on the <code><html></code> element of every eLearning page? ✓ A) So screen readers use the correct pronunciation rules for the page's language B) To enable CSS language selectors C) To comply with SCORM 1.2 standards D) To allow browsers to detect the character encoding Feedback: WCAG 2.1 SC 3.1.1 requires that the default language of each page be programmatically determinable. The <code>lang</code> attribute on <code><html></code> allows screen readers to apply correct speech synthesis rules.	
Q6. A developer writes: <code><input type='text' placeholder='Enter your name'></code>. What WCAG 2.1 failure does this commit? ✓ A) Using placeholder as the only form label fails SC 1.3.5 (Identify Input Purpose) and SC 1.3.1 B) <code>type='text'</code> is not a valid HTML5 attribute C) Placeholders are not allowed in Bootstrap forms D) This code is fully accessible Feedback: Placeholder text disappears when the user starts typing, leaving users without an accessible label. A visible <code><label></code> element is always required.	
Q7. What does <code>box-sizing: border-box</code> do in CSS? ✓ A) Includes padding and border within the element's declared width B) Adds padding and border to the declared width, increasing the total size C) Applies a box shadow to the element D) Sets the element's display to block Feedback: The default CSS box model adds padding and border outside the declared width (content-box). Setting <code>border-box</code> makes the declared width the total element width, preventing layout calculation errors.	
Q8. Which WCAG 2.1 Level AA success criterion sets the minimum contrast ratio for normal text against its background? ✓ A) 1.4.3 Contrast (Minimum) — ratio of 4.5:1 B) 1.4.6 Contrast (Enhanced) — ratio of 7:1 C) 1.4.1 Use of Colour D) 2.4.7 Focus Visible Feedback: SC 1.4.3 requires a minimum contrast ratio of 4.5:1 for normal text at Level AA. SC 1.4.6 requires 7:1 at Level AAA. These apply to text against background colour, not decorative images.	
Q9. In Bootstrap 5, what is the minimum number of column-width units that must be applied to a <code><div></code> within a row for it to span the full width? ✓ A) 12 B) 6 C) 1 D) 100 Feedback: Bootstrap's grid is divided into 12 equal columns. A <code>col-12</code> class spans all 12 units, filling the full row width.	
Q10. [Short Answer] Explain the difference between the CSS Flexbox and Grid layout systems, and give one appropriate use case for each in an eLearning dashboard. Model Answer: Flexbox is a one-dimensional layout system best suited for aligning items along a single axis (row or column). Use case: aligning the navigation bar items horizontally. CSS Grid is a two-dimensional layout system that controls both rows and columns simultaneously. Use case: constructing the overall dashboard layout with a header, sidebar, main content area, and footer arranged in a defined grid.	

End-Unit Assessment 2: HTML/CSS Practical Rubric (Moodle Workshop)

Criterion	Distinction (75–100%)	Merit (60–74%)	Pass (50–59%)	Fail (<50%)
Semantic HTML Structure	All structural elements (<header>, <nav>, <main>, <section>, <footer>) used correctly and appropriately. No unnecessary <div> substitutions for semantic elements.	Most semantic elements used correctly; one or two inappropriate substitutions.	Some semantic elements used; several structural errors or misuse of elements.	Semantic elements absent or largely incorrect; relies entirely on <div>.
WCAG 2.1 Accessibility	All form inputs have associated <label> elements; ARIA attributes correct; focus styles visible; colour contrast ≥4.5:1; no placeholder-only labels.	Most accessibility requirements met; one or two minor failures (e.g., a single unlabelled input).	Partial accessibility implementation; multiple WCAG failures identified.	Accessibility requirements largely unmet; no visible focus styles or form labels.
CSS Styling & Specificity	CSS uses class selectors appropriately; box-sizing: border-box applied; no inline styles; consistent visual style throughout.	Mostly appropriate CSS; one or two inline styles or ID overuse.	CSS present but inconsistent; heavy reliance on inline styles or !important.	Minimal or non-functional CSS; no stylesheet or styles applied.
Responsive Design	Layout fully responsive at all three required breakpoints (mobile, tablet, desktop); no horizontal scrolling at any size.	Layout responds at two of the three breakpoints; minor overflow at one size.	Layout partially responsive; significant issues at mobile or desktop size.	Layout not responsive; fixed-width only.
Code Quality	Code is well-indented, consistently formatted, and includes HTML comments describing key sections.	Mostly well-formatted with minor indentation inconsistencies.	Code functional but poorly formatted; difficult to read.	Code is unformatted or largely non-functional.