College of Contract Management
United Kingdom

Advanced Diploma in

Web Design

Syllabus

Contents

1. Course Structure and Rules of Combination

2. Course Content

1. Course Structure and Rules of Combination

1.1 Rationale

The Advanced Diploma in Web Design course will give students the skills they need to succeed in their career in Web Design. Students will build the competencies they need to become competent and successful Web Designers, teaching them about HTML5, CSS3 Fundamentals, Java Script and DOM Manipulation, Design Tools, Content Management Systems (CMS) and Web Security. This advanced diploma is equivalent to a postgraduate qualification, which will boost the learners' Bachelor's degree and can lead to a rewarding career as a Applications developer, Game developer, Multimedia programmer, Multimedia specialist, SEO specialist, UX designer, UX researcher, Web content manager, Web designer, or Web developer.

This course is delivered via live online lectures. Our course lecturers have significant experience within the industry and will relate lecture content to real-life scenarios. In addition to this, lectures also include both practical examples and case studies. Through this delivery style, learners will be able to reflect on the practical challenges faced by professionals in the industry and establish an understanding of how to act in these situations in a manner that still works towards success.

1.2 Career Progression

The course provides the underpinning knowledge and understanding for the Advanced Diploma in Web Design. It also enables students to study towards a university degree. Upon completion of this course, learners will be able to successful navigate a career in Web Design, either working internally for an employer or completing freelance work.

1.3 Course Rules of Combination

The course can be completed in 2 years (approximately 24 months), and includes an assessment at the end of each module. Each module is worth 10 credits.

- WD101: HTML5 and CSS3 Fundamentals
- WD102: JavaScript and DOM Manipulation
- WD103: User Interface (UI) Design Principles
- WD104: User Experience (UX) Design Principles
- WD105: Advanced Design Tools
- WD106: Advanced Front-End Development
- WD201: Web Accessibility
- WD202: Content Management Systems (CMS)
- WD203: Advanced UI/UX Design
- WD204: Web Security
- WD205: Capstone Project and Final Review

To achieve the Professional Diploma, candidates are required to complete all modules and pass their respective final assessments.

1.4 Entry Requirements

- Minimum 18 years old
- Existing industry experience
- Be able to demonstrate IT competency and basic coding skills

1.5 Module and Assessment Grades

The Assessor will award a grade for the achievement of each module (Fail, Pass, Merit or Distinction). Grades apply to overall performance in modules and assessments.

Indicative marking descriptors for differentiating between levels of achievement when marking assessments are provided below (Section 1.8).

The overall grade for a qualification is calculated using a points system. Each module grade attracts points as follows:

Fail	0 points
Pass	1 point
Merit	2 points
Distinction	3 points
Module Exemption	1 point

1.6 Assessment

The assessment process is set by the College of Contract Management, defining the requirements learners are expected to meet in order to demonstrate that a learning outcome has been achieved. All learning outcomes must be achieved in order to gain attainment of credit for that module.

All completed assessments are marked and verified internally, and are subject to approval by our partner universities or awarding bodies.

The assessment criteria are based on 3 areas:

1. **Task Achievement** - This is a measure of how well the candidate answers the task question(s) and identifies the important aspects of the task.
2. **Technical Content** - This is a measure of how well the candidate identifies, describes and evaluates the technical aspects of the task.
3. **Presentation** - This is a measure of how well the candidate presents the assessment, which includes the quality of the structure and paragraphing, the quality and relevance of visual or graphical content and the referencing used for quoted sources.

1.7 Assessment Policies

1. All submission of an assessment must include:
   a. a copy of the full brief given by the Examination Officer or Course Administrator;
   b. all source material must be cited in the text and a full bibliography of source material (including author, title, publisher, edition and page) listed at the end of the submission.
2. All submissions must be submitted into our system as instructed by the Examination Officer or Course Administrator.
3. All submissions under the student's name must only be the work of that student. All information sources must be acknowledged. There is the **possibility of failing the module if the content of the assessment are deemed be plagiarised** as set out in the rules and regulations of the College.
4. All submissions should be in pdf format (unless software files are specified) and students must keep a copy of all submitted work for reference purposes. Receipt will be acknowledged by the College once the work is submitted via our online exam portal.
5. Whenever a candidate submits work after the approved deadline without an authorised extension, a maximum "Pass" grade will be awarded.
6. The Assessor will comment on the quality of the work for learning purposes.
7. Application for an extension must be requested prior to the submission deadline. Submissions must be made on the exam portal for each module extension request. A primary extension (two weeks) request can be made without the submission of any evidence or reasoning, any further extension requests will require submission of supporting documentation. All requests must be addressed to the Examination Officer or Course Administrator.

1.8 Indicative Marking Descriptors

Note: Please note that the bands below describe indicative characteristics only. An overall holistic approach is required when assessing a candidate's work and assigning a grade. Please read these grading bands in conjunction with the College of Contract Management Assignment Policy.

Grade	Task Achievement - The Relevance of the Response	Inclusion of Relevant Technical Knowledge in Content	Presentation/Coherence
Distinction			
70%+	The work demonstrates a comprehensive understanding of the task. All relevant information is included and the main issues are effectively identified and analysed. There is evaluation and analysis of solutions to any issues relevant to the task. The response shows control of content within the word count.	The work demonstrates a strong understanding of a wide range of technical issues relevant to the task. There is analysis of the advantages/disadvantages of any possible choices, risks and potential outcomes.	The work is appropriately structured and the argument is developed coherently. There is a recognised form of source referencing which supports the points made in the task. Paragraphing and titling are used effectively to assist the reader and the use of visual/graphical information is clear and effective. The graphical information is relevant to the task and is accurate.
Merit			
60-69%	The work demonstrates a clear understanding of the main issues relevant to the task and the issues are explained effectively and any potential solutions are identified. There is an attempt to analyse the merits of the solutions proposed. The task is broadly achieved within the word count and relevant to the assignment.	The work demonstrates an understanding of the key technical issues of the task. There is clear description of relevant technical aspects with some attempt to evaluate the merits of their solutions relating to the set task.	Demonstrates an awareness of presentation and an attempt to present the information with clarity and coherence. There is referencing to any sources used and the paragraphing and titling assists the reader. Clear graphical information supports their assignment and it has broad relevance to the task but, there might be some inaccuracies/omissions in their overall submission.
Pass			
40-59%	The work demonstrates an understanding of the task and the main points of the task are identified. There is no attempt to evaluate or analyse their solutions and there may be some inaccuracies, omissions and irrelevant content within their submission. There may be lack of control in relation to the word count.	The work identifies the main technical issues and demonstrates an understanding of these. Their submission may be limited to a description with little evidence of any evaluation. There may be some omissions and inaccuracies in the detail and there may be some irrelevant information.	There is an attempt to structure the information and evidence of paragraphing and titling has been made but it is not always appropriate. Some basic graphical information may be included which is of some assistance to the reader. There may be some omissions or inaccuracies in their submission and the work is generally coherent however, there may be occasional lapses in coherence and overall structure.
Fail			
0-39%	The work shows a poor understanding of the subject, with frequent inaccuracies and it fails to identify important aspects of the task. Much of the information provided is irrelevant to the task and there may be evidence of copying and pasting from external sources. The response may be limited to lists of words with no attempt to explain the relevance/merits of these to the task set and the assignment falls short of the word count.	The work demonstrates a lack of understanding of the technical aspects and there are omissions of important technical information. Errors are evident in the technical content and there has been no attempt to explain the relevance of the technical content to the task.	The assignment lacks structure and may be limited to lists of points which are not developed, it is also disorganised in structure which makes it difficult for the reader to understand the points being made. The submission is illegible or incoherent in places with no referencing of any external sources used. The graphical illustrations are of poor quality or are absent and they may be irrelevant to the task set. There may be errors and a lack of clarity causing difficulty for the reader to understand.

1.8 Indicative Marking Descriptors

Note: Please note that the bands below describe indicative characteristics only. An overall holistic approach is required when assessing a candidate's work and assigning a grade. Please read these grading bands in conjunction with the College of Contract Management Assignment Policy.

Grade	Task Achievement - The Relevance of the Response	Inclusion of Relevant Technical Knowledge in Content	Presentation/Coherence
Distinction			
80%+	The work demonstrates a comprehensive understanding of the task. All relevant information is included and the main issues are effectively identified and analysed. There is evaluation and analysis of solutions to any issues relevant to the task. The response shows control of content within the word count.	The work demonstrates a strong understanding of a wide range of technical issues relevant to the task. There is analysis of the advantages/disadvantages of any possible choices, risks and potential outcomes.	The work is appropriately structured and the argument is developed coherently. There is a recognised form of source referencing which supports the points made in the task. Paragraphing and titling are used effectively to assist the reader and the use of visual/graphical information is clear and effective. The graphical information is relevant to the task and is accurate.
Merit			
65-79%	The work demonstrates a clear understanding of the main issues relevant to the task and the issues are explained effectively and any potential solutions are identified. There is an attempt to analyse the merits of the solutions proposed. The task is broadly achieved within the word count and relevant to the assignment.	The work demonstrates an understanding of the key technical issues of the task. There is clear description of relevant technical aspects with some attempt to evaluate the merits of their solutions relating to the set task.	Demonstrates an awareness of presentation and an attempt to present the information with clarity and coherence. There is referencing to any sources used and the paragraphing and titling assists the reader. Clear graphical information supports their assignment and it has broad relevance to the task but, there might be some inaccuracies/omissions in their overall submission.
Pass			
55-64%	The work demonstrates an understanding of the task and the main points of the task are identified. There is no attempt to evaluate or analyse their solutions and there may be some inaccuracies, omissions and irrelevant content within their submission. There may be lack of control in relation to the word count.	The work identifies the main technical issues and demonstrates an understanding of these. Their submission may be limited to a description with little evidence of any evaluation. There may be some omissions and inaccuracies in the detail and there may be some irrelevant information.	There is an attempt to structure the information and evidence of paragraphing and titling has been made but it is not always appropriate. Some basic graphical information may be included which is of some assistance to the reader. There may be some omissions or inaccuracies in their submission and the work is generally coherent however, there may be occasional lapses in coherence and overall structure.
Fail			
0-54%	The work shows a poor understanding of the subject, with frequent inaccuracies and it fails to identify important aspects of the task. Much of the information provided is irrelevant to the task and there may be evidence of copying and pasting from external sources. The response may be limited to lists of words with no attempt to explain the relevance/merits of these to the task set and the assignment falls short of the word count.	The work demonstrates a lack of understanding of the technical aspects and there are omissions of important technical information. Errors are evident in the technical content and there has been no attempt to explain the relevance of the technical content to the task.	The assignment lacks structure and may be limited to lists of points which are not developed, it is also disorganised in structure which makes it difficult for the reader to understand the points being made. The submission is illegible or incoherent in places with no referencing of any external sources used. The graphical illustrations are of poor quality or are absent and they may be irrelevant to the task set. There may be errors and a lack of clarity causing difficulty for the reader to understand.

1.9 Calculating Overall Qualification Grade

To calculate the overall qualification grade, the individual module grades should be added together and compared to the table below:

Candidates must pass all 11 modules of the course.

Total Points for all 11 Modules	Overall Grade
33	**Distinction**
32	
31	
30	
29	
28	
27	
26	**Merit**
25	
24	
23	
22	
21	
20	
19	
18	**Pass**
17	
16	
15	
14	
13	
12	
11	
10 or fewer	**Fail**
Candidates must achieve at least a pass in (or hold exemption from) all 11 modules to be awarded the Diploma.	

1.10 Mandatory Modules

Module Reference	Title	Credit Value	LH
WD101	HTML5 and CSS3 Fundamentals	10	100
WD102	JavaScript and DOM Manipulation	10	100
WD103	User Interface (UI) Design Principles	10	100
WD104	User Experience (UX) Design	10	100
WD105	Advanced Design Tools	10	100
WD106	Advanced Front-End Development	10	100
Year 2			
WD201	Web Accessibility	10	100
WD202	Content Management Systems (CMS)	10	100
WD203	Advanced UI/UX Design	10	100
WD204	Web Security	10	100
WD205	Capstone Project and Final Review	10	100

WD101: HTML5 and CSS3 Fundamentals

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Understand HTML5 elements and attributes.	1.1 Use HTML primer. 1.2 Identify new semantic elements. 1.3 Interpret multimedia elements. 1.4 Identify form elements and attributes.
2. Determine HTML forms and input elements.	2.1 Identify form structure and styling. 2.2 Input types and attributes. 2.3 Process form validation.
3. Identify semantic HTML.	3.1 Understand the role of semantic HTML. 3.2 Determine ARIA landmarks and roles. 3.3 Identify SEO best practices.
4. Compare CSS styling techniques.	4.1 Integrate CSS. 4.2 Apply the CSS box model. 4.3 Determine CSS selectors and declarations.
5. Identify advanced CSS selectors.	5.1 Identify combinator selectors. 5.2 Understand attribute selectors. 5.3 Comprehend pseudo selectors.
6. Practice CSS transitions and animations.	6.1 Highlight key transition properties. 6.2 Demonstrate keyframes and animation properties. 6.3 Create smooth animations.
7. Investigate flexbox layout.	7.1 Navigate Flexbox concepts and properties. 7.2 Navigate Flexbox responsive layouts. 7.3 Navigate Flexbox vs. grid.
8. Investigate CSS grid layout.	8.1 Apply grid concepts and properties. 8.2 Determine grid responsive layouts. 8.3 Compare grid vs. flexbox.
9. Navigate CSS frameworks (e.g. Bootstrap, Pure.css).	9.1 Understand CSS frameworks. 9.2 Navigate Bootstrap grid system. 9.3 Demonstrate customising Bootstrap. 9.4 Provide alternative CSS frameworks - e.g. Pure.css.
10. Use responsive images and media.	10.1 Understand responsive techniques. 10.2 Apply responsive images. 10.3 Use responsive media embedding.
11. Interpret CSS pre-processors (e.g. Sass, Less).	11.1 Provide an overview of CSS pre-processors. 11.2 Use variables and mixins. 11.3 Implement nesting and functions in Sass.
12. Project: Building responsive website.	12.1 Applying responsive design principles. 12.2 Integrating CSS frameworks. 12.3 Project review and optimisation.

13. Project: CSS framework implementation.	13.1 Implementing a CSS framework in a project. 13.2 Customising framework components. 13.3 Project view and optimisation.
14. Project: Advanced styling techniques.	14.1 Advanced CSS styling challenges. 14.2 Exploring CSS Grid and Flexbox. 14.3 Refining project styles.
15. Project: Responsive web design project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Final project showcase.

WD102: JavaScript and DOM Manipulation

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Determine JavaScript basics.	1.1 Understand variables, data types, and operators. 1.2 Control flow and looping. 1.3 Determine functions and scope.
2. Identify JavaScript functions and scope.	2.1 Determine function declarations and expressions. 2.2 Identify scope and closures. 2.3 Apply higher-order functions.
3. Understand JavaScript objects and arrays.	3.1 Navigate objects and array in JavaScript. 3.2 Iterate and modify objects. 3.3 Apply array methods.
4. Comprehend DOM manipulation and traversal.	4.1 Understanding the DOM. 4.2 Select and manipulate DOM elements. 4.3 Demonstrate event handling.
5. Demonstrate event handling in JavaScript.	5.1 Identify event types and event listeners. 5.2 Assess event propagation. 5.3 Handle form events.
6. Navigate asynchronous JavaScript (Promises, async/wait).	6.1 Understand asynchronous programming. 6.2 Work with promises. 6.3 Apply Async/Await Syntax.
7. Understand AJAX.	7.1 Provide a clear overview of AJAX. 7.2 Make asynchronous requests. 7.3 Fetch API.
8. Apply JavaScript frameworks (e.g., jQuery).	8.1 Understand JavaScript frameworks. 8.2 Use jQuery selectors and manipulation. 8.3 Apply AJAX with jQuery.
9. Navigate debugging JavaScript.	9.1 Comprehend browser developer tools for debugging. 9.2 Determine console logging and breakpoints. 9.3 Handle common JavaScript errors.
10. Understand cross-browser compatibility.	10.1 Identify cross-browser issues. 10.2 Navigate browser testing tools. 10.3 Apply Polyfills and feature detection.
11. Project: Interactive web elements.	11.1 Build interactive features with JavaScript. 11.2 Integrate DOM manipulation. 11.3 Project review and enhancements.
12. Project: AJAX integration.	12.1 Incorporating AJAX into a project. 12.2 Handling asynchronous requests. 12.3 Project review and optimisation.

13. Project: Cross-browser testing.	13.1 Cross-browser testing strategies. 13.2 Resolving cross-browser compatibility issues. 13.3 Finalising project for review.
14. Project: Debugging and optimisation.	14.1 Optimisation and solutions. 14.2 Clean code techniques. 14.3 Typescript static typing.
15. Project: JavaScript project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Final project showcase.

WD103: User Interface (UI) Design Principles

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Understand UI design principles.	1.1 Identify fundamentals UI principles. 1.2 Provide consistency and feedback. 1.3 Understand the hierarchy and balance in UI.
2. Explore colour theory and application.	2.1 Understand colour psychology. 2.2 Apply colour harmony and schemes. 2.3 Apply colour in UI designs.
3. Comprehend typography in UI design.	3.1 Navigate typeface selection. 3.2 Understand font pairing and hierarchy. 3.3 Manage readability and accessibility.
4. Manage layout and composition.	4.1 Use grid systems in UI design. 4.2 Understand visual hierarchy and alignment. 4.3 Demonstrate responsive layouts.
5. Apply iconography and imagery.	5.1 Comprehend icon design principles. 5.2 Manage image selection and optimisation. 5.3 Apply icon fonts and SVGs.
6. Select UI design patterns.	6.1 Identify common UI design patterns. 6.2 Apply navigation and interaction patterns. 6.3 Supply consistency across patterns.
7. Comprehend designing for different devices.	7.1 Understand responsive design strategies. 7.2 Mobile-first design approach. 7.3 Adapting UI for various screens.
8. Demonstrate UI prototyping.	8.1 Understand prototyping tools overview. 8.2 Create low-fidelity prototypes. 8.3 Demonstrate interactive prototyping techniques.
9. Navigate design systems.	9.1 Understand design systems. 9.2 Identify components and style guides. 9.3 Maintain a design system.
10. Understand UI animation principles.	10.1 Identify the purpose of UI animations. 10.2 Acknowledge types of UI animations. 10.3 Use animation tools and techniques.
11. Comprehend microinteractions.	11.1 Understand their definition and importance. 11.2 Create microinteractions. 11.3 Identify microinteraction patterns.
12. Use UI design tools (e.g. Sketch, Figma).	12.1 Comprehend UI design tools. 12.2 Identify key features and capabilities. 12.3 Apply collaborative design in tools.

13. Project: UI design exploration.	13.1 Exploring UI design concepts. 13.2 Sketching and ideation. 13.3 Initial UI design concepts.
14. Project: UI prototyping.	14.1 Translating UI designs to prototypes. 14.2 Interactivity in prototypes. 14.3 Iterative prototyping.
15. Project: Design system implementation.	15.1 Building a design system. 15.2 Applying design patterns. 15.3 Finalising UI design project.

WD104: User Experience Design

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Define User Experience (UX).	1.1 Interpret the definition and importance of UX. 1.2 Assess the user-centered design approach. 1.3 Compare UX vs. UI design.
2. Assess user research methods.	2.1 Compare qualitative vs. quantitative research. 2.2 Interpret user interviews and surveys. 2.3 Monitor user personas.
3. Create user personas.	3.1 Practice developing detailed personas. 3.2 Understand persona empathy mapping. 3.3 Manage persona validation.
4. Understand user journey mapping.	4.1 Monitor mapping user interactions. 4.2 Identify pain points and opportunities. 4.3 Assess iterative user journey mapping.
5. Explore information architecture.	5.1 Organise information structure. 5.2 Navigate site maps and hierarchies. 5.3 Understand navigation design.
6. Comprehend card sorting and user flows.	6.1 Conduct card sorting exercises. 6.2 Create user flows. 6.3 Analyse user flow metrics.
7. Practice wireframing techniques.	7.1 Compare low-fidelity vs. high-fidelity wireframes. 7.2 Assess the wireframing tools overview. 7.3 Practice iterative wireframing.
8. Explore prototyping for UX.	8.1 Determine prototyping importance in UX. 8.2 Manage interactive prototypes. 8.3 Explore user testing prototypes.
9. Partake in usability testing.	9.1 Plan and conduct usability tests. 9.2 Gather and analysing feedback. 9.3 Complete iterative testing.
10. Monitor the iterative design process.	10.1 Define iteration in UX. 10.2 Demonstrate continuous improvement. 10.3 Integrate feedback.
11. Manage UX design tools.	11.1 Define UX design tools. 11.2 Use collaborative UX design platforms. 11.3 Manage prototyping and testing tools.
12. Project: UX design process overview.	12.1 Documenting UX design process. 12.2 Project planning and scope. 12.3 Initial user research

13. Project: Wireframing and prototyping.	13.1 Translating research to wireframes. 13.2 Developing high-fidelity prototypes. 13.3 Iterative prototyping.
14. Project: Usability testing.	14.1 Planning and conducting usability tests. 14.2 Analysing usability metrics. 14.3 Refining prototypes based on feedback.
15. Project: UX design project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Final UX design showcase.

WD105: Advanced Design Tools

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Understand advanced use of design tools (e.g. Adobe XD).	1.1 Understand advanced features in design tools. 1.2 Integrate design components. 1.3 Monitor collaborative design workflows.
2. Identify collaboration features in design tools.	2.1 Comprehend multi-user editing and collaboration. 2.2 Interpret version control and design history. 2.3 Integrate design tools with development.
3. Integrate design tools with development.	3.1 Manage handoff to developers. 3.2 Export assets for development. 3.3 Bridge the gap between design and development.
4. Design handoff to developers.	4.1 Demonstrate best practices for handling off designs. 4.2 Design specifications and documentation. 4.3 Communicate with development teams.
5. Design versioning and history.	5.1 Test version control in design tools. 5.2 Track design changes. 5.3 Revert to previous versions.
6. Design for different screen sizes.	6.1 Demonstrate responsive design in design tools. 6.2 Preview designs across devices. 6.3 Adapt layouts for various screens.
7. Manage responsive design in design tools.	7.1 Monitor responsive artboards and canvases. 7.2 Implement breakpoint strategies. 7.3 Design for mobile-first.
8. Develop advanced prototyping techniques.	8.1 Report on advanced interactivity in prototypes. 8.2 Understand conditional interactions. 8.3 Simulate real-world scenarios.
9. Design for accessibility.	9.1 Integrate accessibility features in designs. 9.2 Understand accessibility testing in design tools. 9.3 Design inclusive interfaces.
10. Project: Advanced design tool usage.	10.1 Apply advanced features in a project. 10.2 Demonstrate multi-user collaboration. 10.3 Manage iterative design refinement.
11. Project: Collaboration with developers.	11.1 Provide effective collaboration strategies. 11.2 Oversee the handoff process in a collaborative environment. 11.3 Address development challenges.
12. Project: Design handoff and iterations.	12.1 Handoff documentation. 12.2 Addressing developer feedback. 12.3 Iterative design enhancements.

13. Project: Responsive design implementation.	13.1 Implementing responsive design principles. 13.2 Adapting designs for different devices. 13.3 Responsive design testing.
14. Project: Design accessibility audit.	14.1 Evaluating designs for accessibility. 14.2 Implementing accessibility recommendations. 14.3 Accessibility testing and verification.
15. Project: Design tool mastery and review.	15.1 Final collaborative project review. 15.2 Mastery of design tools. 15.3 Showcasing advanced design skills.

WD106: Advanced Front-End Development

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Understand advanced CSS frameworks (e.g. Tailwind CSS).	1.1 Define advanced CSS frameworks. 1.2 Understand tailwind CSS concepts and usage. 1.3 Customise tailwind styles.
2. Navigate front-end build tools (e.g. Webpack).	2.1 Determine the role of build tools in front-end development. 2.2 Explore Webpack. 2.3 Configure and optimise Webpack.
3. Explore CSS-in-JS libraries.	3.1 Identify the benefits of CSS-in-JS. 3.2 Interpret popular CSS-in-JS Libraries (e.g. styled-components). 3.3 Test implementation in front-end projects.
4. Comprehend front-end frameworks (e.g. React, Angular, Vue.js).	4.1 Define front-end frameworks. 4.2 Choose the right framework for the project. 4.3 Setting up a front-end framework project.
5. Assess component-based architecture.	5.1 Understand components. 5.2 Build reusable components. 5.3 Assess component state and props.
6. Monitor state management in front-end applications.	6.1 Identify the importance of state management. 6.2 Global state management libraries. 6.3 Implementing state management in a project.
7. Determine server-side rendering vs. client-side rendering.	7.1 Compare server-side and client-side rendering. 7.2 Assess the benefits and challenges of each approach. 7.3 Choose the right rendering strategy.
8. Explore Progressive Web Apps (PWAs).	8.1 Identify what a PWA is. 8.2 Build PWAs with front-end frameworks. 8.3 Provide offline support and background sync.
9. Use WebAssembly and its applications.	9.1 Define WebAssembly 9.2 Use cases and advantages. 9.3 Integrating WebAssembly in front-end projects.
10. Practice web performance optimisation techniques.	10.1 Identify strategies for optimising front-end performance. 10.2 Explore lazy loading and code splitting. 10.3 Measure and analyse web performance.
11. Project: Implementing advanced front-end techniques.	11.1 Apply advanced front-end concepts. 11.2 Integrate front-end frameworks. 11.3 Addressing performance challenges.
12. Project: Building a front-end application with a framework.	12.1 Developing a real-world front-end application. 12.2 Integrating state management. 12.3 Addressing performance challenges.

13. Project: State management in a front-end application.	13.1 Implementing global state management. 13.2 Testing and debugging state. 13.3 State management optimisation.
14. Project: Optimising web performance.	14.1 Analysing and optimising project performance. 14.2 Implementing web performance best practices. 14.3 Finalising front-end project for review.
15. Project: Advanced front-end project review.	14.1 Collaborative project review. 14.2 Iterative feedback and improvement. 14.3 Showcasing advanced front-end skills.

WD201: Web Accessibility

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Understand Web Accessibility Guidelines (WCAG).	1.1 Define WCAG. 1.2 Understand structure and principles of WCAG. 1.3 Follow compliance levels (A, AA, AAA).
2. Create accessible design practices.	2.1 Design for keyboard accessibility. 2.2 Provide meaningful text alternatives. 2.3 Demonstrate colour contrast for readability.
3. Comprehend ARIA (Accessible Rich Internet Applications) roles.	3.1 Define ARIA. 3.2 Manage ARIA roles and attributes. 3.3 Enhance accessibility with ARIA.
4. Use assistive technologies.	4.1 Define assistive technologies. 4.2 Provide screen readers and VoiceOvers. 4.3 Demonstrate testing with assistive technologies.
5. Navigate testing tools for accessibility.	5.1 Practice automated accessibility testing. 5.2 Demonstrate manual accessibility testing. 5.3 Use accessible auditing tools.
6. Design for cognitive disabilities.	6.1 Identify cognitive accessibility challenges. 6.2 Provide inclusive design strategies. 6.3 Demonstrate user testing with cognitive disabilities.
7. Design for mobility disabilities.	7.1 Identify challenges faced by users with mobility disabilities. 7.2 Adapt navigation for accessibility. 7.3 Demonstrate user testing with mobility disabilities.
8. Design for visual impairments.	8.1 Comprehend visual impairments. 8.2 Implement text-to-speech and screen magnifications. 8.3 Demonstrate testing with visual impairments.
9. Demonstrate inclusive design principles.	9.1 Determine principles of inclusive design. 9.2 Design for diverse user needs. 9.3 Explore inclusive design case studies.
10. Demonstrate accessibility in multimedia content.	10.1 Provide captioning and transcription for videos. 10.2 Describe visual content for accessibility. 10.3 Test multimedia accessibility.
11. Project: Accessibility audit.	11.1 Conduct accessibility audits. 11.2 Identify and address accessibility issues. 11.3 Iterative accessibility improvements.
12. Project: Implementing accessibility features.	12.1 Integrating accessibility features in a project. 12.2 User testing and diverse audiences. 12.3 Finalising accessibility features.

13. Project: User testing with diverse audiences.	13.1 Planning and conducting user tests. 13.2 Gathering feedback from diverse users. 13.3 Iterative design based on test results.
14. Project: Inclusive design implementation.	14.1 Applying inclusive design principles. 14.2 Addressing specific user needs. 14.3 Finalising inclusive design features.
15. Project: Accessibility project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Showcase of accessible design features.

WD202: Content Management Systems (CMS)

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Identify content management systems.	1.1 Determine the definition and purpose of CMS. 1.2 Identify types of CMS (Traditional vs. Headless). 1.3 List the pros and cons of using a CMS.
2. Navigate popular CMS platforms (e.g. WordPress, Joomla, Drupal).	2.1 Understand WordPress, Joomla, and Drupal. 2.2 Choose the right CMS for a project. 2.3 Install and setup a CMS.
3. Understand customising and theming CMS.	3.1 Customise CMS templates and themes. 3.2 Create custom page templates. 3.3 Determine theming best practices.
4. Learn CMS security best practices.	4.1 Understand common CMS security risks. 4.2 Apply security best practices. 4.3 Manage regular security audits.
5. Manage headless CMS concepts.	5.1 Answer: What is a headless CMS? 5.2 List the benefits and drawbacks of headless CMS. 5.3 Integrate a headless CMS in design.
6. Integrate CMS with web design.	6.1 Design with CMS content in mind. 6.2 Create dynamic content. 6.3 Leverage CMS features in design.
7. Use CMS plugins and extensions.	7.1 Define CMS plugins and extensions. 7.2 Evaluate and install plugins. 7.3 Demonstrate custom plugin development.
8. Understand multi-language support in CMS.	8.1 Implement multilingual websites. 8.2 Understand language switching and URL structure. 8.3 Comprehend translation and localisation.
9. Demonstrate E-commerce integration with CMS.	9.1 Understand E-commerce capabilities in CMS. 9.2 Integrate payment gateways. 9.3 Highlight security considerations in E-commerce.
10. Project: Setting up a CMS website.	10.1 Choosing and installing a CMS. 10.2 Configuring basic settings. 10.3 Initial content population.
11. Project: Customising CMS themes.	11.1 Customising CMS templates and themes. 11.2 Implementing custom functionality. 11.3 Initial content population.
12. Project: Securing a CMS website.	12.1 Implementing security measures. 12.2 Regular security audits. 12.3 Monitoring and responding to security threats.

13. Project: Headless CMS implementation.	13.1 Integrating a headless CMS in web design. 13.2 Designing for a headless architecture. 13.3 Content retrieval and display.
14. Project: E-commerce integration.	14.1 Adding E-commerce functionality to a website. 14.2 Integrating payment gateways. 14.3 Addressing e-commerce security concerns.
15. Project: CMS project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Showcase of CMS website features.

WD203: Advanced UI/UX Design

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Demonstrate advanced UI design principles.	1.1 Understand gestalt principles in UI. 1.2 Design for emotion. 1.3 Create intuitive user interfaces.
2. Comprehend advanced colour theory in UI design.	2.1 Determine the psychological impact of colours. 2.2 Create harmonious colour schemes. 2.3 Assess colour accessibility and contrast.
3. Understand advanced typography in UI design.	3.1 Use custom fonts and variable fonts. 3.2 Experiment with expressive typography. 3.3 Apply typography animation techniques.
4. Identify motion design in UI.	4.1 Identify the purpose and principles of motion design. 4.2 Interpret microinteractions with motion. 4.3 Demonstrate storytelling through animation.
5. Understand prototyping with advanced interactions.	5.1 Understand prototyping tools for advanced interactions. 5.2 Apply complex user flows in prototypes. 5.3 Use interactive transitions and states.
6. Design for emerging technologies (VR, AR).	6.1 Define VR and AR. 6.2 Address UI/UX challenges in immersive design. 6.3 Create immersive experiences.
7. Demonstrate advanced user research methods.	7.1 Conduct ethnographic research. 7.2 Provide diary studies and longitudinal research. 7.3 Demonstrate advanced usability testing techniques.
8. Design for internationalisation.	8.1 Adapt design for global audiences. 8.2 Understand multilingual UI design. 8.3 Appreciate cultural considerations in design.
9. Design for emotional engagement.	9.1 Provide emotional design strategies. 9.2 Create memorable user experiences. 9.3 Measure emotion engagement.
10. Comprehend UX writing and microcopy.	10.1 Understand the importance of UX writing. 10.2 Craft effective microcopy. 10.3 Display inclusive language in UX writing.
11. Project: Advanced UI design exploration.	11.1 Exploring innovative UI design concepts. 11.2 Applying advanced design principles. 11.3 Iterative design refinement.
12. Project: Motion design in UI.	12.1 Integrating motion design in a project. 12.2 Enhancing user experience with animation. 12.3 Motion design iterations.

13. Project: Prototyping complex interactions.	13.1 Translating complex interaction to prototypes. 13.2 Advanced prototyping tools and techniques. 13.3 User testing with interactive prototypes.
14. Project: Designing for emerging technologies.	14.1 Designing for virtual and augmented reality. 14.2 Creating immersive UI/UX. 14.3 Iterative design for emerging tech.
15. Project: Advanced UI/UX design project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Showcase of advanced UI/UX design.

WD204: Web Security

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Define web security.	1.1 Define web security. 1.2 Identify key security threats and risks. 1.3 View security as a continuous process.
2. Identify common web security vulnerabilities.	2.1 Understand SQL injection. 2.2 Navigate Cross-Site Scripting (XSS). 2.3 Navigate Cross-Site Request Forgery (CSRF). 2.4 Identify security best practices for each vulnerability.
3. Implement HTTPS and SSL/TLS.	3.1 Understand SSL/TLS. 3.2 Implement HTTPS. 3.3 SSL/TLS certificates and configuration.
4. Explore Cross-Site Scripting (XSS) prevention.	4.1 Input validation and sanitisation. 4.2 Determine a content security policy (CSP). 4.3 Manage secure coding practices.
5. Understand Cross-Site Request Forgery (CSRF) prevention.	5.1 Identify synchroniser token pattern. 5.2 Use Anti-CSRF tokens. 5.3 Demonstrate double-sumit cookies.
6. Navigate SQL injection prevention.	6.1 Provide parameterised statements. 6.2 Identify stored procedures. 6.3 Manage ORM (Object-Relational Mapping) usage.
7. Demonstrate security best practices in front-end development.	7.1 Identify client-side security considerations. 7.2 Provide secure data transmission. 7.3 List client-side authentication best practices.
8. Demonstrate security best practices in back-end development.	8.1 Demonstrate input validation and sanitisation. 8.2 Provide session management. 8.3 Secure communication with databases.
9. Configure Web Application Firewalls (WAF).	9.1 Understand the role of WAF in web security. 9.2 Configure and manage a WAF. 9.3 Demonstrate WAF best practices.
10. Complete security audits and penetration testing.	10.1 Conduct security audits. 10.2 Plan and execute penetration tests. 10.3 Manage bug bounty programs.
11. Project: Implementing web security best practices.	11.1 Apply security best practices in a project. 11.2 Secure coding and configuration. 11.3 Iterative security improvements.
12. Project: Security audit and recommendations.	12.1 Conducting a security audit. 12.2 Documenting security vulnerabilities. 12.3 Providing security recommendations.

13. Project: Penetration testing.	13.1 Planning and executing penetration tests. 13.2 Addressing security weaknesses. 13.3 Finalising project security.
14. Project: Securing a web application.	14.1 Implementing comprehensive security measures. 14.2 Continuous monitoring and updating. 14.3 Final security assessment.
15. Project: Web security project review.	15.1 Collaborative project review. 15.2 Iterative feedback and improvement. 15.3 Showcase of secure web application.

WD205: Capstone Project and Final Review

Learning outcomes: The learner will	Assessment criteria: The learner can
1. Capstone Project kickoff.	1.1 Understand the Capstone Project. 1.2 Determine the project scope and objectives. 1.3 Choose a real-world problem or client.
2. Complete project planning and documentation.	2.1 Create a project plan. 2.2 Define deliverables and milestones. 2.3 Demonstrate document standards.
3. Undertake design thinking in the Capstone Project.	3.1 Apply design thinking principles. 3.2 Complete user-centric problem solving. 3.3 Demonstrate iterative prototyping and testing.
4. Navigate the development phase.	4.1 Implement front-end and back-end functionality. 4.2 Demonstrate version control and collaboration. 4.3 Practice continuous integration and deployment.
5. Complete user testing feedback.	5.1 Conduct user tests. 5.2 Gather user feedback. 5.3 Understand iterative design and development.
6. Finalise the Capstone Project.	6.1 Refine features based on feedback. 6.2 Display performance optimisation. 6.3 Complete final testing a quality assurance.
7. Complete project documentation and reporting.	7.1 Create comprehensive documentation. 7.2 Complete technical reports and user manuals. 7.3 Present project findings and outcomes.
8. Demonstrate professional development skills.	8.1 Build a professional portfolio. 8.2 Create a resume and cover letter. 8.3 Create job search strategies.
9. Complete interview preparation.	9.1 Demonstrate behavioural and technical interview techniques. 9.2 Practice portfolio presentation in interviews. 9.3 Handle common interview questions.
10. Practice networking and industry engagement.	10.1 Build a professional network. 10.2 Participate in industry events. 10.3 Navigate online presence and social media.
11. Explore freelancing and entrepreneurship.	11.1 Explore freelancing opportunities. 11.2 Start a web design business. 11.3 Undertake client management and contracts.
12. Demonstrate continued learning and skill enhancement.	12.1 Stay updated with industry trends. 12.2 Complete online courses and certifications. 12.3 Join professional organisations.

13. Consider ethics and professionalism in web design.	13.1 Demonstrate ethical considerations in web design. 13.2 Show professional conduct and integrity. 13.3 Identify responsibilities to clients and users.
14. Complete final project presentation.	14.1 Deliver a professional presentation. 14.2 Showcase Capstone Project highlights. 14.3 Handle questions and feedback.
15. Course reflection and graduation.	15.1 Reflect on the learning journey. 15.2 Graduation ceremony and certificates. 15.3 Identify next steps in professional development.