MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Web Design and Programming**			**Module Delivery**	
Module Type	**Core**			☒ **Theory**	
Module Code	**IT3101**			☒ **Lab**	
ECTS Credits	**6**				
SWL (hr/sem)	**150**				
Module Level	UGIII		**Semester of Delivery**	1	
Administering Department	Information Technology		**College**	College of Computer Science and Information Technology	
Module Leader	Nabil Sadiq Al-Sharifi		**e-mail**	nabeel@uowa.edu.iq	
Module Leader's Acad. Title	Assistant Lecturer		**Module Leader's Qualification**	MSc	
Module Tutor	Nabil Sadiq Al-Sharifi		**e-mail**	nabeel@uowa.edu.iq	
Peer Reviewer Name	Dr. Ali Karim Abdulrahim		**e-mail**	alialmujab@uowa.edu.iq	
Scientific Committee Approval Date	01/09/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	The aim of the Web Technologies module is to provide students with a comprehensive understanding of key web development concepts, tools, and techniques. Students will learn front-end skills, including HTML, CSS, and JavaScript, to design visually appealing and interactive web interfaces. They will also gain knowledge about essential web technologies such as TCP/IP, sessions, and cookies. The module emphasizes industry best practices, focusing on code organization, performance optimization, and accessibility considerations. Additionally, a final project, developed in collaboration with group members, serves as a demonstration of the skills and knowledge acquired throughout the module. By the end of the module, students will be proficient in designing and building engaging web interfaces and will be equipped to apply their learning in real-world web development scenarios.
Module Learning Outcomes	1. Module Learning Outcomes for Web Technologies (Front-End Focus): 2. Demonstrate an understanding of the fundamental concepts of web technologies and their significance in modern web development. 3. Apply HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) to create visually appealing and well-structured web pages. 4. Utilize JavaScript to implement interactive and dynamic functionality on web pages, including form validation, event handling, and DOM manipulation. 5. Explain the basic principles of the TCP/IP protocol and understand how it enables communication over the internet. 6. Implement session management techniques, such as using cookies, to maintain user state and personalize web experiences. 7. Design user-friendly and responsive web interfaces that adapt to different devices and screen sizes using responsive design techniques. 8. Apply best practices in front-end development, including code organization, accessibility considerations, and cross-browser compatibility. 9. Understand the importance of web performance optimization techniques, including minimizing file sizes, leveraging caching, and reducing page load times. 10. By achieving these learning outcomes, students will develop a solid foundation in front-end web development skills, enabling them to create engaging and functional web interfaces while adhering to industry best practices.
Indicative Contents	Indicative content for the Web Technologies subject includes the following: ● Introduction to Web Technologies This section provides a comprehensive introduction to web technologies. It covers the fundamental concepts and principles that underpin the World Wide Web, including the client-server architecture, request-response model, and the role of web browsers. Additionally, it explores the evolution of web technologies, from static HTML web pages to dynamic web applications, and discusses the importance of standards and protocols in web development. ● Front-End Development

	Front-end development focuses on designing and building the user interface of web applications. This topic covers essential skills and tools used in front-end development, such as HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript. It includes a discussion on creating responsive and accessible web interfaces, as well as incorporating interactivity and multimedia elements into web pages. ● Web Development Tools and Workflow This section focuses on the various tools and workflows used in modern web development. It introduces integrated development environments (IDEs), code editors, version control systems, and task runners. It also covers topics such as debugging techniques, browser developer tools, and deployment strategies for web applications. By covering these key areas in web technologies, students will gain a solid foundation in both the theory and practical aspects of web development, enabling them to design and build effective web applications.

Learning and Teaching Strategies	
Strategies	In teaching web technologies, effective learning and teaching strategies are crucial for providing students with a comprehensive understanding of the subject. A solid theoretical foundation is essential, encompassing the history, protocols, and standards that have shaped web technology. This knowledge contextualizes the significance of design and development practices. To reinforce learning, hands-on strategies such as practical exercises, coding challenges, and real-world examples engage students and allow them to apply their theoretical knowledge. Active participation through group projects and individual assignments fosters skill development and problem-solving abilities. Regular feedback and guidance ensure students are well-prepared to tackle real-world web development scenarios confidently. By combining theory with practical application, students gain the necessary technical skills and critical thinking abilities to excel in designing and building the front end of web applications.

Student Workload (SWL)			
Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	4	2 (8%)	2,4,8,10	1,2,3,4,5,6,7
	Project	1	5 (5%)	12	all
	Lab	5	3 (15%)	3,5,7,9,11	all
	Homework	4	2 (8%)	2,5,9,12	all
	Onsite Assignments	2	2 (8%)	5,10	all
Summative assessment	Midterm Exam	1	%10	7	
	Final Exam	1	%50	16	
Total assessment			%100		

Delivery Plan (Weekly Syllabus)	
	Material Covered
Week 1	Introduction about WWW
Week 2	Internet and Domain Name System
Week 3	World Wide Web and HTTP
Week 4	Web Documents Types
Week 5	Cookie, session, OSI Layer
Week 6	HTML I
Week 7	HTML II
Week 8	Final Project Orientation Session
Week 9	Using Table and DIV in HTML
Week 10	Cascading Style Sheets
Week 11	Page layout and HTML forms
Week 12	JavaScript fundamentals
Week 13	JavaScript Functions and Forms
Week 14	Event Handlers and Document Objects
Week 15	Introduction of Front-End Framework
Week 16	Final Exam

Delivery Plan (Weekly Lab. Syllabus)	
	Material Covered
Week 1	HTML Structure and Tags Lab: Hands-on lab to practice creating the basic structure of an HTML document, understanding the role of HTML tags, and organizing content using headings, paragraphs, lists, and semantic elements.
Week 2	CSS Styling Lab: Practical exercises to apply CSS rules and properties for styling web page elements, including fonts, colors, backgrounds, borders, and box model properties.
Week 3	CSS Layout Lab: Lab session focusing on creating responsive layouts using CSS grid and flexbox, experimenting with different column arrangements, alignment, and responsive breakpoints.
Week 4	Media Queries Lab: Interactive lab to explore the implementation of media queries in CSS, designing responsive designs that adapt to different devices, screen sizes, and orientations.
Week 5	JavaScript Basics Lab: Hands-on exercises to reinforce understanding of JavaScript fundamentals, including variables, data types, operators, conditionals, loops, and functions.
Week 6	DOM Manipulation Lab: Practical lab activities to manipulate the DOM using JavaScript, dynamically updating content, handling events, and modifying element properties.
Week 7	JavaScript Event Handling Lab: Lab session to practice event handling techniques, attaching event listeners, and responding to user interactions, such as button clicks, form submissions, and mouse movements.
Week 8	Form Validation Lab: Interactive lab exercises to implement form validation using JavaScript, ensuring data integrity and providing user feedback for input errors.
Week 9	Responsive Web Design Lab: Lab activities focused on designing and building responsive web pages using CSS media queries and flexible layouts, testing the responsiveness across different devices and screen sizes.
Week 10	JavaScript Libraries Lab: Hands-on lab session to explore popular front-end JavaScript libraries or frameworks.
Week 11	CSS Preprocessors Lab: Practical exercises using CSS preprocessors (e.g., Sass or Less) to enhance the CSS workflow, including creating variables, mixins, and nesting rules for more maintainable and modular stylesheets.
Week 12	Web Accessibility Lab: Lab session dedicated to implementing web accessibility best practices, including using semantic HTML, adding alternative text to images, and testing accessibility features using screen readers.
Week 13	Web Performance Optimization Lab: Lab activities to optimize web page performance, including minifying CSS and JavaScript files, optimizing image sizes, and leveraging caching techniques for faster loading times.
Week 14	Div-based Layout Lab: Hands-on lab to explore the concept of div-based layouts in HTML and CSS. Students will practice creating multi-column layouts, using div elements and CSS properties such as float, clear, and positioning to achieve different page structures and designs.

Week 15	Table-based Layout Lab: Lab session focusing on creating page layouts using HTML tables. Students will learn how to structure content using table elements, apply CSS styles to customize table appearance, and understand the advantages and limitations of table-based layouts in modern web development.
Week 16	Final Exam

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	● HTML for the World Wide Web, Fifth Edition, with XHTML and CSS by Elizabeth Castro ● JavaScript A Beginner's Guide Fourth Edition John Pollock	
Recommended Texts		
Websites	http://www.w3schools.com	

APPENDIX:

GRADING SCHEME				
Group	**Grade**	**Mark**	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	Fail	(0-44)	Considerable amount of work required
Note:				

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.