MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Web programming**			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☒ **Practical** ☐ **Seminar**	
Module Code	TUCS				
ECTS Credits	6				
SWL (hr/sem)	**150**				
Module Level		1	**Semester of Delivery**		2nd
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Yahya Layth Khaleel		**e-mail**	yahya@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		master
Module Tutor			**e-mail**		
Peer Reviewer Name		Harith Abdullah	**e-mail**		
Scientific Committee Approval Date		07/06/2023	**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture. 2- Develop skills in analyzing the usability of a web site. 3- Understand how to plan and conduct user research related to web usability. 4- Learn the core web technologies and programming languages that power the modern web. Starting with HTML and CSS and Javascript. 5- Learn some concepts in server-side programming using (PHP), along with MySQL database. 6- - Exploring how to create dynamic web applications that can process user data, store information, using content management systems (CMS).
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Understand the fundamental concepts of web development: Gain a solid understanding of how the web works, including client-server architecture, HTTP protocols, and the role of web browsers. 2. Develop proficiency in HTML and CSS: Acquire the skills to create well-structured web pages using HTML markup, apply CSS styles for layout and design, and create responsive and visually appealing websites. 3. Understand JavaScript programming: Learn the essentials of JavaScript programming, including variables, data types, and operators. 4. Build dynamic web applications: Explore server-side programming using PHP. Understand how to process user data, connect to databases, and generate dynamic content. 5. Work with databases: Gain familiarity with database management systems like MySQL. Learn how to design and create database schemas. 6. Implement security measures: Understand common web security vulnerabilities and learn techniques to protect web applications from attacks. 7. Problem-solving and debugging skills: Develop the ability to identify and fix common issues in web development through effective

	troubleshooting and debugging techniques.
Indicative Contents المحتويات الإرشادية	1- Introduction to Web Development • Overview of web technologies and standards • Client-server architecture and HTTP protocols • Understanding web browsers and their rendering engines • HTML Fundamentals 2- Structure of HTML • HTML tags and elements • Working with text, images, links, and lists • Creating forms for user input • Semantic HTML and accessibility best practices 3- CSS Styling • CSS syntax and selectors • Box model and layout techniques • Applying styles to text, colors, backgrounds, and borders • CSS positioning and responsive design • CSS frameworks and libraries 4- JavaScript Basics • Introduction to JavaScript and its role in web development • Variables, data types, and operators 5- Server-Side Programming • Introduction to server-side programming languages (PHP) • Handling user input and form data • Working with databases (MySQL) • Templates for dynamic content • Session management and user authentication 6-Web Security • Common web security vulnerabilities • Guidelines and directions to protect the website

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Hands-on Practice: This Encourage students to actively engage in hands-on coding exercises and projects. Providing them with opportunities to

	apply theoretical concepts in practical scenarios, allowing them to gain proficiency through practice. 2. Project-Based Learning: Assigning projects that simulate real-world web development scenarios. This approach allows students to apply their knowledge and skills to create fully functional web applications, reinforcing their understanding and problem-solving abilities. 3. Code Review and Feedback: Incorporating code review sessions where students can share their code and receive constructive feedback. This process helps students identify areas for improvement, learn best practices, and enhance their coding style and techniques. 4. Collaborative Learning: Foster a collaborative learning environment where students can work together on group projects or problem-solving tasks. Encourage peer-to-peer discussions, code sharing, and knowledge exchange, as this can enhance understanding and expose students to diverse perspectives and solutions. 5. Online Resources and Documentation: Introduce students to reputable online resources, documentation, and tutorials related to web development. Teaching students how to effectively search for solutions, read and understand documentation, and leverage online communities and forums for support and learning. 6. Practical Examples and Case Studies: Provide practical examples and case studies that demonstrate the application of web programming concepts in real-world scenarios. This helps students relate theoretical concepts to practical use cases, enhancing their understanding and problem-solving abilities. 7. Regular Assessments and Feedback: Conducting regular assessments, quizzes, and coding challenges to evaluate students' progress and understanding. Providing timely feedback to help students identify their strengths and areas that require improvement. 8. Continuous Learning and Exploration: Encouragement students to stay updated with the latest trends, tools, and technologies in web development. 9. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	77	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	73	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.8
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	2	10% (10)	3, 12	
	Projects	1	10% (10)	8-14	
	Report				
Summative assessment	**Midterm Exam**	2 hr	20% (20)	11	
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction web programming
Week 2	Internet and Intranet, Web application ,web page, website , Classifying websites
Week 3	Client side script and server side scrip, Introduction to HTML
Week 4	HTML tags and attributes
Week 5	HTML - Titles and headings
Week 6	HTML – Lists
Week 7	HTML Images
Week 8	HTML – Tables
Week 9	HTML Frame and Form
Week 10	Introduction to CSS
Week 11	Internal and external CSS

Week 12	Introduction to JavaScript
Week 13	JavaScript Variables, data types, and operators
Week 14	Server-Side Programming (PHP with MySQL)
Week 15	Content Management Systems (CMS), Security vulnerabilities

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	**Material Covered**
Week 1	**HTML tags and attributes**
Week 2	**HTML – Titles, headings and Lists**
Week 3	**HTML Images**
Week 4	**HTML – Tables**
Week 5	**HTML Frame and Form**
Week 6	**Internal CSS**
Week 7	**External CSS**
Week 8	**Creating web page structure using HTML tags**
Week 9	**JavaScript variables definitions, data types**
Week 10	**JavaScript operators**
Week 11	**Implementing interactivity and dynamic content on web pages**
Week 12	**Install WAMP server**
Week 13	**Work with MySQL database**
Week 14	**Content Management Systems (CMS): install Joomla**
Week 15	**Create dynamic website**

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	None	
Recommended Texts	Jon, Duckett. "HTML and CSS: Design and Build Websites." (2016).	
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

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.