Web Programming (CUE31033)

1. Information on the Programme

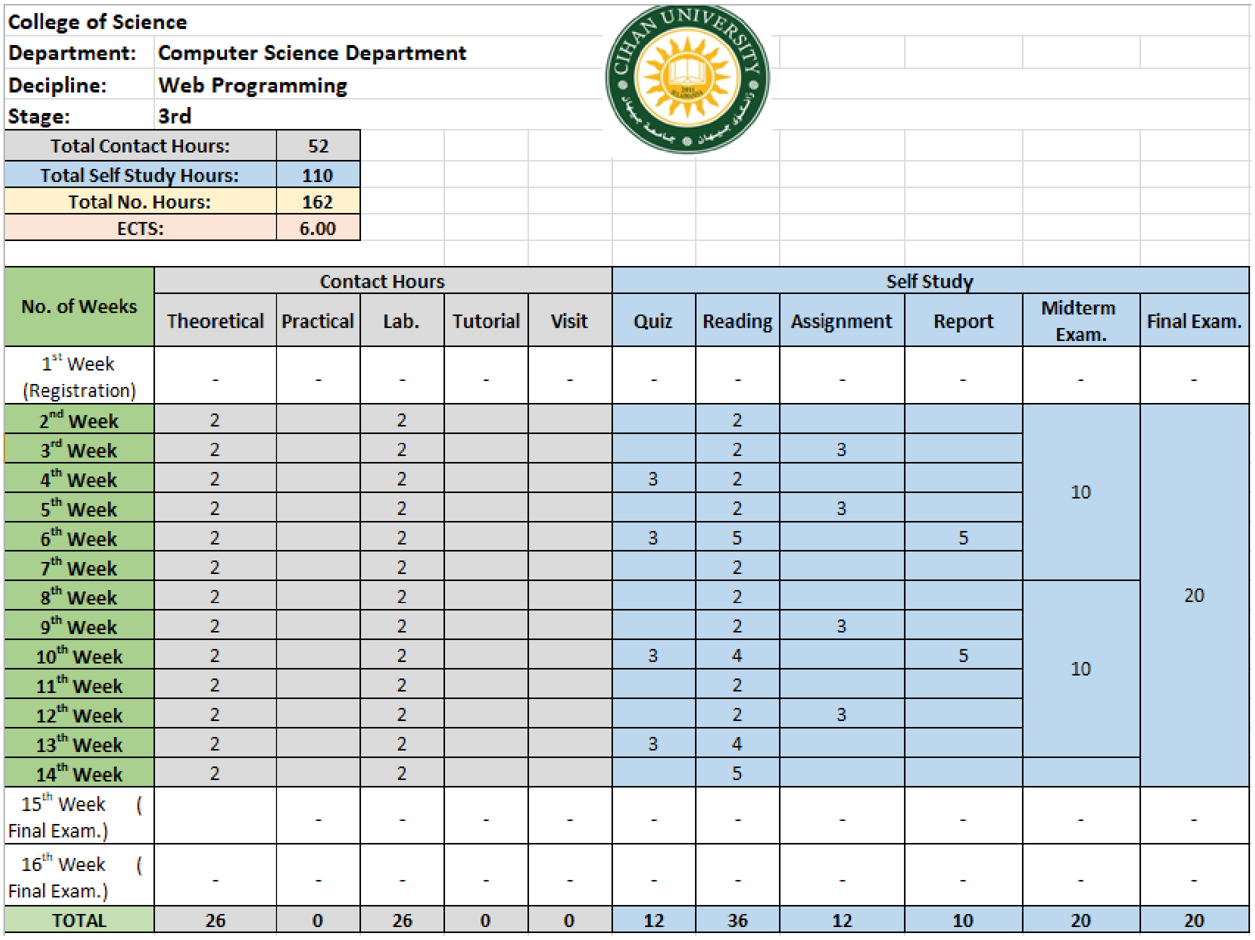
1.1. Higher Education Institution	Cihan University Sulaimaniya
1.2. College	Science
1.3. Department	Computer Science
1.4. Field of Study	Web Programming
1.5. Cycle of Study[1]	1
1.6. Specialization/ Study Programme	Computer Science
1.7. Form of Education	Full Time

2. Information on the Discipline

2.1. Discipline Name	Web Programming
2.2. Code	CUE31043
2.3. Language:	English
2.4. (Theory) Lecturer **E-mail:** **Tel:** **Webpage, Google Classroom**	Dr. Asan Baker Kanbar asan.baker@sulicihan.edu.krd 07702396919
2.5. Practical/Seminar/ Laboratory/ Project Lecturer **e-mail:** **Tel:** **Webpage, Google Classroom**	Dr. Asan Baker Kanbar asan.baker@sulicihan.edu.krd 07702396919

2.6. Year of Study	-2022 2023	2.7 .Semester	1st	2.8. Assessment Type[2]	Written exam, & CE	2.9. Discipline Status	Content[3]	CD
							Mandotary[4]	MD

3. Total estimated time (Teaching Hours per Semester)



College of Science

Department:	**Computer Science Department**
Decipline:	**Web Programming**
Stage:	**3rd**

Total Contact Hours:	52
Total Self Study Hours:	110
Total No. Hours:	162
ECTS:	6.00

No. of Weeks	Contact Hours					Self Study					
	Theoretical	Practical	Lab.	Tutorial	Visit	Quiz	Reading	Assignment	Report	Midterm Exam.	Final Exam.
1st Week (Registration)	-	-	-	-	-	-	-	-	-	-	-
2nd Week	2		2				2			10	20
3rd Week	2		2				2	3			
4th Week	2		2			3	2				
5th Week	2		2				2	3			
6th Week	2		2			3	5		5		
7th Week	2		2				2				
8th Week	2		2				2			10	
9th Week	2		2				2	3			
10th Week	2		2			3	4		5		
11th Week	2		2				2				
12th Week	2		2				2	3			
13th Week	2		2			3	4				
14th Week	2		2				5				
15th Week (Final Exam.)		-	-	-	-	-	-	-	-	-	-
16th Week (Final Exam.)	-	-	-	-	-	-	-	-	-	-	-
TOTAL	26	0	26	0	0	12	36	12	10	20	20

4. Prerequisites (if applicable)

4.1 Curriculum-Related	The curriculum focuses on web programming languages (such as HTML, CSS, JavaScript, and server-side scripting),
4.2 Skills-Related	Acquire proficiency in web programming skills, including HTML, CSS, JavaScript, server-side scripting (e.g., PHP, Python, or ASP.NET),

5. Conditions (if applicable)

5.1. For the Theoretical	1. Read and comprehend the textbook material. 2. Attend all the classes and take notes on class discussions. 3. Actively participate in class discussions and activities. 4. Submit all the assignments and the project on time. 5. Pass tests and quizzes.
5.2. For the Practical	All students are normally required to attend the Lab; take part in lectures through applying the exercises on the computer or as quizzes, and to implement projects.

6. Cumulated Specific Competences

Professional Competencies	▪ Proficiency in Programming Languages: A strong foundation in programming languages such as HTML, CSS, JavaScript, and server-side languages (e.g., PHP, Python, Ruby) is crucial for web programming. Understanding the syntax, concepts, and best practices of these languages is essential. ▪ Web Development Frameworks and Libraries: Familiarity with popular web development frameworks and libraries, such as React, Angular, Vue.js, Express.js, and Django, can greatly enhance productivity and simplify the development process. ▪ Front-End Development: Competence in front-end technologies like HTML, CSS, and JavaScript is essential for creating user interfaces, designing responsive layouts, and implementing interactive features on the client side. ▪ Back-End Development: Proficiency in server-side programming languages, such as PHP, Python, or Node.js, and knowledge of databases (e.g., MySQL, MongoDB) are important for handling server-side logic, data storage, and retrieval. ▪ API Integration: Understanding how to integrate with external APIs (Application Programming Interfaces) to fetch and process data from third-party services is a valuable skill for web programming. ▪ Security and Performance Optimization: Knowledge of security best practices, such as secure authentication and data encryption, is crucial to protect web applications from vulnerabilities. Additionally, skills in optimizing web performance, including caching strategies, minimizing file sizes, and reducing server requests, are important for delivering fast and efficient websites.
Transversal competences	• Problem Solving: The ability to analyze complex problems, identify efficient solutions, and troubleshoot issues that arise during the development process is crucial in web programming. • Critical Thinking: Web programmers need to think critically to evaluate different approaches, assess the feasibility of solutions, and make informed decisions when designing and implementing web applications. • Attention to Detail: Paying attention to detail is important in web programming to ensure accurate coding, identify and fix errors, and deliver high-quality web applications. • Time Management: Web programmers often work on multiple projects and have deadlines to meet. Effective time management skills help in prioritizing tasks, meeting project milestones, and delivering projects on time. • Adaptability and Flexibility: The field of web programming is constantly evolving, with new technologies and frameworks emerging. Being adaptable and open to learning new tools and techniques is essential for staying current and relevant in the industry.

7. Discipline Objectives (Based on the cumulated specific Competences)

7.1. General Objective	• Application Development: The primary objective of web programming is to build web applications that perform specific functions or provide services to users. This involves designing and implementing the functionality and features of the application using programming languages, frameworks, and technologies. • User Experience (UX): A key objective of web programming is to create a positive and seamless user experience for website visitors. This includes ensuring intuitive navigation, responsive design, fast loading times, and interactive elements that engage and satisfy users. • Dynamic Content: Web programming enables the creation of websites that can dynamically generate and display content based on user interactions or data retrieved from databases or external sources. The objective is to deliver personalized and up-to-date content to users. • Data Management: Web programming involves handling and managing data within web applications. This includes storing, retrieving, updating, and deleting data from databases or other data sources. The objective is to ensure accurate data management and efficient data processing. • Integration of Third-Party Services: Web programming often involves integrating web applications with external APIs (Application Programming Interfaces) and services to enhance functionality. The objective is to seamlessly connect the web application with external services and retrieve or exchange data as needed. • Security: An important objective of web programming is to implement security measures to protect web applications and user data. This includes ensuring secure data transmission, implementing authentication and authorization mechanisms, and protecting against common security vulnerabilities. • Performance Optimization: Web programming aims to optimize the performance of web applications by minimizing loading times, reducing server requests, and optimizing code and assets. The objective is to deliver fast and efficient web experiences to users. • Compatibility and Cross-Browser Support: Web programming strives to ensure that web applications are compatible with different web browsers and devices. The objective is to create web applications that work consistently and provide a consistent user experience across various platforms. • Maintenance and Upgrades: Web programming involves ongoing maintenance and updates to web applications. The objective is to continuously improve and enhance the functionality, performance, and security of the web application as technologies and requirements evolve. • Collaboration and Teamwork: Web programming often involves working as part of a development team, collaborating with designers, project managers, and other programmers. The objective is to effectively communicate and work together to deliver high-quality web applications. • These objectives provide a general framework for web programming, and the specific objectives may vary depending on the nature of the project, the technologies used, and the requirements of the web application being developed. •
7.2. Specific Objectives	• Write clean and semantically correct HTML and CSS code. • Implement interactivity and dynamic behavior using JavaScript. • Develop server-side logic and handle data processing on the server.

	• Integrate external APIs for data retrieval and exchange. • Ensure security measures and implement user authentication. • Optimize web application performance for faster loading.

8. Content

week	8.1. Theoretical-Number of Hours	Teaching methods	Observation
1	registration		
2	• Introduction to Web Development and ASP.NET Core MVC • Overview of web development • Introduction to ASP.NET Core MVC • Setting up the development environment	lecture	**1 lecture = 2 hours**
3	HTML and CSS Fundamentals • Structure of an HTML document • Styling with CSS • Responsive design basics	lecture, assignment	**1 lecture = 2 hours**
4	• JavaScript Essentials • Introduction to JavaScript and its role in web development • Manipulating the DOM • Basic event handling	lecture, Quiz	**1 lecture = 2 hours**
5	• Introduction to C# and .NET Core • Basics of C# programming • Overview of .NET Core • Writing simple C# code in a console application	lecture,	**1 lecture = 2 hours**
6	• Understanding MVC Architecture • Detailed exploration of the Model-View-Controller pattern • Creating a simple MVC application • How routing works in ASP.NET Core MVC	Lecture (report1)	**1 lecture = 2 hours**
7	**MIDTERM EXAM 1**		
8	• Creating Views in ASP.NET Core MVC • Razor syntax for creating dynamic HTML • Using layouts and partial views • Introduction to ViewData and ViewBag	Lecture	**1 lecture = 2 hours**

9	Advanced Razor Syntax and View Components • Razor directives and expressions • Conditional rendering and loops in Razor • Creating and using View Components for reusable UI	lecture, Assignment	**1 lecture = 2 hours**
10	Working with Forms in ASP.NET Core MVC • Building and rendering HTML forms • Understanding model binding (without a database) • Validating user input in forms •	Lecture,Quiz (report2)	**1 lecture = 2 hours**
11	**MIDTERM EXAM 2**		
12	CSS Frameworks: Bootstrap Integration • Introduction to Bootstrap and responsive design • Integrating Bootstrap with ASP.NET Core MVC • Customizing Bootstrap for your application	lecture,	**1 lecture = 2 hours**
13	• Enhancing forms with JavaScript • Handling client-side events • Simple form validation with JavaScript	lecture,	**1 lecture = 2 hours**
14	Working with Partial Views and View Components • Creating partial views for modular design • Using View Components to manage complex UI logic • Best practices for reusable UI components	lecture	**1 lecture = 2 hours**
15	• Week 14: Putting It All Together: Building a Small Project • Planning and designing a small web project • Implementing MVC principles in the project • Applying Bootstrap and Razor techniques	lecture	**1 lecture = 2 hours**

week	**8.2. Practical Works–Number of Hours**	**Teaching methods**	**Observation**
1	registration		
2	Based on lecture 1	Lecture	**1lecture = 2 hours**

3	Based on lecture 2	Lecture,	**1 lecture = 2 hours**
4	Based on lecture 3	Lecture	**1 lecture = 2 hours**
5	Based on lecture 4	Lecture, assignment	**1 lecture = 2 hours**
6	Based on lecture 5	Lecture,Quiz	**1 lecture = 2 hours**
7	**MIDTERM EXAM 1**		**2 hours**
8	Based on lecture 6	Lecture	**1 lecture = 2 hours**
9	Based on lecture 7	Lecture,	**1 lecture = 2 hours**
10	Based on lecture 8	Lecture	**1 lecture = 2 hours**
11	**MIDTERM EXAM 2**		**2 hours**
12	Based on lecture 9	Lecture, Assignment	**1 lecture = 2 hours**
13	Based on lecture 10	Lecture,Quiz	**1 lecture = 2 hours**
14	Based on lecture 11	Lecture,	**1 lecture = 2 hours**
15	Based on lecture 12		

- ***Compulsory Bibliography:***

Key references:

HTML and CSS: Design and Build Websites by Jon Duckett (2011)
Responsive Web Design by Ethan Marcotte (2011)
"The Principles of Beautiful Web Design by Jason Beaird and James George (2014)

ional Bibliography:

9. Assessment

Type of Activity	9.1. Assessment Criteria[2]	9.2. Assessment Type	9.3. Percentage of the final Grade
9.4. Theoretical	Mid-term (20%) Final –exam (35%)	Exam	**%55**
9.5. Practical/ Seminar/Laboratory	Mid-term (10%) Final-Exam (15%)	Exam	**%25**
9.6. Activity during Semester	Quizzes (10%) + Assignment(10%)+	Exam	**%20**

Minimum performance Standards:

Theoretical Lecturer	**Dr. Asan Baker Kanbar**
Practice Lecturer	**Dr. Asan Baker Kanbar**

Approved by the Curriculum development Committee:	
1	
2	
3	
Head of the Department	**Dr. Asan Baker Kanbar**