

SYLLABUS

Web Programming

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Computer Science
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	Web Programming			Discipline code	MLE5015		
2.2. Course coordinator	Assoc.prof.phd Adrian Sterca						
2.3. Seminar coordinator	Assoc.prof.phd Adrian Sterca						
2.4. Year of study	2	2.5. Semester	4	2.6. Type of evaluation	E	2.7. Discipline regime	Mandatory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	4	of which: 3.2 course	2	3.3 seminar/laboratory/project	2
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	28
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					20
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					30
Tutorship					9
Evaluations					15
Other activities:					0
3.7. Total individual study hours					94
3.8. Total hours per semester					150
3.9. Number of ECTS credits					6

4. Prerequisites (if necessary)

4.1. curriculum	Computer Networks, Distributed Operating Systems, Databases, Data Structures and Algorithms, Object Oriented, Programming
4.2. competencies	Elementary knowledge on working with an SQL database server, fundamental knowledge about the structure of the Internet and the way the Internet functions, basic knowledge on data structures and algorithms, programming languages, object-oriented programming.

5. Conditions (if necessary)

5.1. for the course	Class room with a video projector device
5.2. for the seminar /lab activities	

6.1. Specific competencies acquired ¹

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	<ul style="list-style-type: none"> • advanced programming skills in high-level programming languages • development and maintenance of software systems
Transversal competencies	<ul style="list-style-type: none"> • Applying rules for an organized and efficient work, responsible attitude towards the didactic-scientific field for creative capitalization of one's own potential, complying to the principles and professional ethics norms. • Utilizing efficient methods and techniques for learning, knowing, research and development of knowledge capitalization capacities, adapting to the requirements of a dynamic society and the communication in Romanian or an international language.

6.2. Learning outcomes

Knowledge	<ul style="list-style-type: none"> • The graduate has the necessary knowledge for using computers, developing software programs and applications, information processing. • The graduate knows multiple programming languages and is able to write applications in compiled, interpreted or dynamic languages with the ability to choose the appropriate programming language for the specific application to be developed.
Skills	<ul style="list-style-type: none"> • The graduate has the ability to develop, design and create new applications, systems or products using best practices of the field. • The graduate has the ability to understand and use design patterns for application development. • The graduate has the basic knowledge required to install, configure and maintain a server system on the Internet.
Responsibility and autonomy:	<ul style="list-style-type: none"> • The graduate has the necessary knowledge to process and verify data and information. • The graduate has the ability to observe and obtain information from various sources.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	<ul style="list-style-type: none"> • To introduce students to modern techniques for web programming using both server-side and client-side technologies. The course is meant as an introductory course in web technologies.
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • Understanding how the World Wide Web is built and functions • Knowing the main technologies/languages used in web development: HTML/XML, CSS, Javascript/DOM, PHP, Angular, React, JSP/Servlet, ASP.NET Core

8. Content

8.1 Course	Teaching methods	Remarks
1. WWW history and concepts: The Internet addressing mechanism, name servers, URLs	Exposure:description, explanation,examples,	

and URIs	discussion of case studies	
2. HTML – HyperText Markup Language. HTML 5	Exposure:description, explanation,examples, discussion of case studies	
3. HTTP – HyperText Transfer Protocol	Exposure:description, explanation,examples, discussion of case studies	
4. CSS – Cascading Style Sheets. CSS3. Responsive design. Web fonts and icons. CSS preprocessors.	Exposure:description, explanation,examples, discussion of case studies	
5. XML languages. XHTML, XML, XSLT	Exposure:description, explanation,examples, discussion of case studies	
6. DOM – Document Object Model. The Javascript language: fundamental concepts, functions, objects, collections, async programming (setTimeout, promises). Javascript browser API.	Exposure:description, explanation,examples, discussion of case studies	
7. Javascript libraries: jQuery	Exposure:description, explanation,examples, discussion of case studies	
8. Javascript frameworks: Angular ; React	Exposure:description, explanation,examples, discussion of case studies	
9. Server-side technologies: PHP. JSON – Javascript Object Notation, AJAX/Fetch API	Exposure:description, explanation,examples, discussion of case studies	
10. Server-side technologies: JSP and Java servlets, Springboot. Hibernate. Rest APIs. API documentation (Swagger)	Exposure:description, explanation,examples, discussion of case studies	
11. Server-side technologies: ASP .NET Core. MVC projects. Entity framework. REST API projects.	Exposure:description, explanation,examples, discussion of case studies	
12. Other web technologies: graphics (WebGL), real-time communication (WebRTC).	Exposure:description, explanation,examples, discussion of case studies	
13. Web security: same-origin principle, cross-site scripting, sql injection.	Exposure:description, explanation,examples, discussion of case studies	
14. Web security: same-origin principle, cross-site scripting, sql injection.	Exposure:description, explanation,examples, discussion of case studies	
Bibliography		
1. http://www.cs.ubbcluj.ro/~forest/wp		
2. A. Sterca, Web programming textbook, https://www.cs.ubbcluj.ro/~forest/wp/courses/		
3. W3Schools Online Web Tutorials, http://www.w3schools.com		
4. http://www.php.net		
5. https://angular.dev/		
6. https://react.dev/		
7. https://spring.io/projects/spring-boot		
8. https://www.baeldung.com/spring-boot		
9. https://dotnet.microsoft.com/en-us/apps/aspnet		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Laboratory work: using HTML 5 main tags	Dialogue, debate, case studies, examples	
2. Laboratory work: CSS tasks	Dialogue, debate, case studies, examples	
3. Laboratory work: CSS layouts	Dialogue, debate, case studies, examples	

4. Laboratory work: XML and XSLT	Dialogue, debate, case studies, examples	
5. Laboratory work: Javascript and DOM (DHTML)	Dialogue, debate, case studies, examples	
6. Laboratory work: jQuery	Dialogue, debate, case studies, examples	
7. Laboratory work: AJAX and PHP	Dialogue, debate, case studies, examples	
8. Laboratory work: Angular	Dialogue, debate, case studies, examples	
9. Laboratory work: Java servlets and JSP	Dialogue, debate, case studies, examples	
10. Laboratory work: Java servlets and JSP	Dialogue, debate, case studies, examples	
11. Laboratory work: Asp .Net Core	Dialogue, debate, case studies, examples	
12. Laboratory work: Asp .Net Core	Dialogue, debate, case studies, examples	
13. Students deliver the last laboratory tasks. Preparing the final exam.	Dialogue, debate, case studies, examples	
14. Students deliver the last laboratory tasks. Preparing the final exam.	Dialogue, debate, case studies, examples	
Bibliography 1. http://www.cs.ubbcluj.ro/~forest/wp 2. A. Sterca, Web programming textbook, https://www.cs.ubbcluj.ro/~forest/wp/courses/ 3. W3Schools Online Web Tutorials, http://www.w3schools.com 4. http://www.php.net 5. https://angular.dev/ 6. https://react.dev/ 7. https://spring.io/projects/spring-boot 8. https://www.baeldung.com/spring-boot 9. https://dotnet.microsoft.com/en-us/apps/aspnet		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying programs of all major universities in Romania and abroad;
- The content of the course is considered by software companies as important for average programming skills

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Knowing the theoretical issues discussed during the course. Being able to solve small practical problems similar to the ones students get during the laboratory activity.	Practical exam	60%

10.5 Seminar/laboratory	Applying the knowledge received from the course. Students get in each laboratory class a task they need to solve.	The lab mark is the average of the marks the student gets on the laboratory work performed by him/her during the semester.	40%
10.6 Minimum standard of performance			
<ul style="list-style-type: none"> In order to successfully pass this class, the practical exam mark and the laboratory mark must be at least 5. Also, the student must participate to at least 90% of the laboratories and at least 50% of the courses. The course requirements are described at: http://www.cs.ubbcluj.ro/~forest/wp 			

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

Date:

...

Signature of course coordinator

Assoc.prof.phd. Adrian STERCA

Signature of seminar coordinator

Assoc.prof.phd. Adrian STERCA

Date of approval:

...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

² Keep only the labels that, according to the [Procedure for applying ODD labels in the academic process](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.