



DISCIPLINE ACADEMIC SHEET
ACADEMIC YEAR 2025 – 2026

1. PROGRAMME DATA

1.1. Higher Education Institution	UNIVERSITY OF CRAIOVA
1.2. Faculty	Automation, Computers and Electronics
1.3. Department	Computers and Information Technology
1.4. Field of Study	Computers and Information Technology
1.5. Study level	L(License / Undergraduate)
1.6. Organizational form	IF
1.7. Study Program (name/code) /Qualification	Computers / L2060101010

2. DISCIPLINE DATA

2.1 Discipline Name	WEB Technologies						
2.2 Course Activities Holder	Prof. dr. ing. Marius MARIAN						
2.3. Practical Activities Holder							
2.4. Study Year	4	2.5. Semester	2	2.6. Evaluation type	E	2.7. Discipline regime	DFA

3. ESTIMATED TOTAL TIME (hours per semester of teaching activities)

3.1. Number of hours per week	3	in which: 3.2 course	2	3.3. sem./lab./pr.	1
3.4. Total hours of curriculum	30	in which: 3.5 course	20	3.6. sem./lab./pr.	10
3.7. Time distribution - hours/week					hours
Study after manual, course support, bibliography and notes					15
Additional documentation in library, on specialized electronic platforms and field					10
Training seminars / labs, homework, portfolios and essays					15
Tutorial					-
Examinations					3
Other activities.....					2
Total hours per individual activities					45
3.8. Total hours per semester					75
3.9. Number of credits					3

4. PRECONDITIONS (where appropriate)

4.1. of curriculum	Students must possess specialized knowledge acquired in the following disciplines: Computer Programming and Programming Languages, Analysis and Synthesis of Digital Devices, Databases, Operating Systems and Real-Time Languages, Object-Oriented Programming.
4.2. of competence	Not necessary.

5. CONDITIONS (where appropriate)

5.1. of the course	The course is taught online / using the video projector. The board is used for some explanations and answers to questions from the audience. Course support is provided in electronic format
5.2. of seminar/ laboratory/project	

6. COURSE OBJECTIVES – expected learning outcomes that are supported by completing and passing the course

Knowledge	The student/graduate identifies and summarizes elementary concepts and methods regarding programming languages, programming environments, programming techniques, databases, artificial intelligence and software engineering and the way they are applied to concrete problems.
Skills (Abilities)	The student/graduate: selects and explains concepts specific to algorithm design, object-oriented programming, logic and functional programming. The student/graduate specifies requirements, analyzes, elaborates, develops and tests software systems in general-purpose or object-oriented programming languages, applying the elements specific to software engineering. The student/graduate designs, implements, develops or manages applications that include various types of databases. The student understands the main concepts underlying translation systems and source code compilation. Contributes to the development and improvement of tools specific to programming languages.
Responsibility and autonomy	The student/graduate behaves honorably, responsibly and ethically, in the spirit of the law, to ensure the reputation of the profession.

7. CONTENT

7.1. Course	The method of implementation	Teaching methods	Allocated time (hours)
1. Introduction. What the concept of dynamic web content represents; a quick introduction to the HTTP protocol and the HTML language; the major benefits of the PHP, MySQL and JavaScript technologies; the Apache web server; the open source paradigm; the characteristics of integrated web application development environments of the WAMP, LAMP, XAMP type		The course is taught online / using the video projector. Practical exemplification of the presented concepts and	2
2. The elements of the PHP scripting language. Embedding PHP code and HTML code; the general structure of a program written in PHP; the syntax of the language; comments; variables; operators (precedence, associativity, relational operators); constants; functions, expressions (literals and			2

variables); decision structures (the if, else, elseif, switch statements, the ? operator); iteration structures (the while, do..while, for loops, the break and continue statements); implicit and explicit variable conversion; dynamic binding in PHP			
3. Functions, objects and arrays in PHP. PHP functions (definition, returning a value, returning an array, returning global variables, passing a reference, the scope of a variable); including support files; PHP objects (declaring a class, creating an object, accessing an object, constructors, methods, properties, constants, the scope of properties and methods); PHP arrays (numerically indexed arrays, associative arrays, using the array keyword, the foreach..as loop, multi-dimensional arrays, standard array manipulation functions)			2
4. Form processing and other PHP-specific functions. Creating forms; the GET and POST methods of the HTTP protocol; retrieving data transmitted through forms; display functions; time and date functions; file manipulation functions; system call functions; using cookies (creation and placement; access; destruction); using sessions (creation, termination, security); HTTP authentication			2
5. Introduction to MySQL. Basic elements; database-specific terminology; the concept of database management system; accessing the MySQL DBMS (through the command line, summary of useful commands, summary of data types); the importance of indexes (creation, querying a MySQL database, table joins, using logical operators); specific MySQL functions			2
6. Elements of database design and administration. Relational databases; primary keys; normalization (the three normal forms); types of relationships (one to one, one to many, many to many); transactions; database backup and recovery			2
7. Accessing MySQL databases via PHP. Querying a MySQL database using PHP; the \$_POST array; practical			2

exemplification			
8. Introduction to JavaScript. Basic elements (writing specific code, placing JavaScript code in HTML/PHP code); comments; variables (string, numeric, arrays, global/local); operators (arithmetic, logical, assignment, relational, etc.); functions; using the DOM - Document Object Model; using expressions (literals and variables); operators (precedence, associativity, relational); the with statement; error/exception handling using onError and try..catch; decision structures (if, switch, ?); iteration structures (while, do..while, for, break, continue)			2
9. Functions, objects, arrays in JavaScript. Functions (definition, returning a value or an array); objects (declaring a class, an object, accessing an object); arrays (numeric, associative, multi-dimensional, using array-specific methods)			2
10. Validation with PHP and JavaScript, error handling, using AJAX. Validating user-entered data with JavaScript; regular expressions; what the AJAX framework represents; using AJAX.			2
Bibliography:			
1. R. Nixon, Learning PHP, MySQL and Javascript, O'Reilly, 2009, ISBN 978-0-596-15713-5 2. E. Hall, Internet Core Protocols: The Definitive Guide, O'Reilly, 2000, ISBN: 1-56592-572-6 3. D. Flanagan, JavaScript: The Definitive Guide, O'Reilly, 4th edition, 2004, ISBN: 0-596-00048-0 4. B. Kennedy, C. Musciano, HTML & XHTML: The Definitive Guide, O'Reilly, 5th edition, 2002, ISBN: 0-596-00382-X 5. P. Hudson, PHP in a nutshell, O'Reilly, 2005, ISBN: 0-596-10067-1 6. E.R. Harold, W.S. Means, XML in a nutshell, O'Reilly, 2004, ISBN: 0-596-00764-7 7. E. Van der Vlist, XML Schema, O'Reilly, 2002, ISBN: 0-596-00252-1			
1.			

7.2. Seminar/laboratory/project	The method of implementation	Teaching methods	Allocated time (hours)
Laboratory 1. The concept of dynamic web content; a quick introduction to the HTTP protocol and the HTML language; the major benefits of the PHP, MySQL and JavaScript technologies; the Apache web server; the open source paradigm; the characteristics of integrated web application development environments of the WAMP, LAMP, XAMP type		The laboratory works are carried out using web scripting/programming languages. Students are provided with laboratory platforms containing a theoretical summary and the way of carrying out the work.	2

Laboratory 2. Embedding PHP code and HTML code; the general structure of a program written in PHP; the syntax of the language; comments; variables; operators (precedence, associativity, relational operators); constants; functions, expressions (literals and variables); decision structures (the if, else, elseif, switch statements, the ? operator); iteration structures (the while, do..while, for loops, the break and continue statements); implicit and explicit variable conversion; dynamic binding in PHP. Laboratory 3. PHP functions (definition, returning a value, returning an array, returning global variables, passing a reference, the scope of a variable); including support files; PHP objects (declaring a class, creating an object, accessing an object, constructors, methods, properties, constants, the scope of properties and methods); PHP arrays (numerically indexed arrays, associative arrays, using the array keyword, the foreach..as loop, multi-dimensional arrays, standard array manipulation functions)		Activities: ■ 70% carrying out the work; ■ 30% interpretation of the results and discussions with the students.	1
Laboratory 4. Creating forms; the GET and POST methods of the HTTP protocol; retrieving data transmitted through forms; display functions; time and date functions; file manipulation functions; system call functions; using cookies (creation and placement; access; destruction); using sessions (creation, termination, security); HTTP authentication			1
Laboratory 5. Basic elements; database-specific terminology; the concept of database management system; accessing the MySQL DBMS (through the command line, summary of useful commands, summary of data types); the importance of indexes (creation, querying a MySQL database, table joins, using logical operators); specific MySQL functions			1
Laboratory 6. Relational databases;			1

primary keys; normalization (the three normal forms); types of relationships (one to one, one to many, many to many); transactions; database backup and recovery			
Laboratory 7. Querying a MySQL database using PHP; the \$_POST array; practical exemplification			1
Laboratory 8. Basic elements (writing specific code, placing JavaScript code in HTML/PHP code); comments; variables (string, numeric, arrays, global/local); operators (arithmetic, logical, assignment, relational, etc.); functions; using the DOM - Document Object Model; using expressions (literals and variables); operators (precedence, associativity, relational); the with statement; error/exception handling using onError and try..catch; decision structures (if, switch, ?); iteration structures (while, do..while, for, break, continue);			1
Laboratory 9. Functions (definition, returning a value or an array); objects (declaring a class, an object, accessing an object); arrays (numeric, associative, multi-dimensional, using array-specific methods)			1
Laboratory 10. Validating user-entered data with JavaScript; regular expressions; using AJAX			1
Bibliography:			
1. R. Nixon, Learning PHP, MySQL and Javascript, editura O'Reilly, 2009, ISBN 978-0-596- 15713-5 2. E. Hall, Internet Core Protocols: The Definitive Guide, editura O'Reilly, 2000, ISBN : 1- 56592-572-6 3. D. Flanagan, JavaScript: The Definitive Guide, editura O'Reilly, 4th edition, 2004, ISBN: 0-596-00048-0 4. B. Kennedy, C. Musciano, HTML & XHTML: The Definitive Guide, editura O'Reilly, 5th edition, 2002, ISBN: 0-596-00382-X 5. P. Hudson, PHP in a nutshell, editura O'Reilly, 2005, ISBN: 0-596-10067-1 6. E.R. Harold, W.S. Means, XML in a nutshell, editura O'Reilly, 2004, ISBN: 0-596-00764-7 7. E. Van der Vlist, XML Schema, editura O'Reilly, 2002, ISBN: 0-596-00252-1			

8. Correlating the content of the discipline with the expectations of representatives from the epistemic community, professional associations and representative employers in the field related to the program

The discipline content was discussed with the representatives of: ▪ S.C. C-S România S.A. Craiova ▪ S.C. NetRom Software S.R.L. Craiova

9. EVALUATION

9. EVALUATION			
Activity Type	9.1. Evaluation criteria	9.2. Evaluation methods	9.3. Final mark weight
9.4. Course	• Understanding the theoretical fundamentals of web application programming • The ability to solve most types of proposed problems using the studied technologies • The ability to make connections between the taught notions • The capacity for analysis and synthesis in a concrete situation	Final written exam / online multiple choice;	60%
9.5. Seminar/laboratory/project	Laboratory: • The ability to analyze, identify the optimal solution and solve the proposed problems • Correct solving of the proposed homework • Correct application of the principles of object-oriented programming • The solutions of the applications are presented and discussed within the group	Ongoing assessment of the student's individual performance and their solving of all homework assignments	40%
9.6. Minimum standard of performance			
▪ Obtaining at least 50% of the score of the ongoing assessments, laboratory tests and final exam; ▪ The final grade is calculated by rounding the final score to a whole grade.			

Date of Completion	Signature of the Course Activities Holder,	Signature of the Practical Activities Holder,
23.08.2025	Prof. dr. ing. Marius MARIAN	

Date of Approval in the Department	Signature of the Department Director,
25.08.2025	Ș.l.dr.ing. Nicolae ENESCU

Date of Approval in the Faculty	Signature of the Dean,
	Prof.dr.ing. Dorin ȘENDRESCU