

KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED
Dean of IT Faculty
HOVORUSHCHENKO T.
2025.

WORKING PROGRAMME OF THE EDUCATIONAL COMPONENT Web Technologies

Field of Study: F Informational technology
Specialty: F2 « Software Engineering»
Level of Higher Education: First (Bachelor's) Level
Educational and Professional Programme: Software Engineering
Course Load: 4 ECTS credits
Course Code: CPT.11
Language of Instruction: Ukrainian
Status of the Educational Component: Compulsory (Professional Training)
Faculty: Faculty of Information Technology
Department: Department of Software Engineering

Form of Study	Year	Semester	Total Credits		Number of hours						Course project	Coursework	Semester control form	
					Contact Hours					Independent Work (incl. Individual Tasks)			pass/ fail test	Exam
			ECTS credits	hours	Total	Lectures	Laboratory works	Practical classes	Seminar classes					
D	3	5	4	120	66	32	34			54	+			+

The working programme is based on the Educational and Professional Programme “Software Engineering” within the specialty 121 “Software Engineering”.

Program's author  Oksana YASHYNA

Approved at the meeting of the Department of Software Engineering

Minutes No. 1 dated August 28, 2025

Head of the Department

 Leonid Bedratyuk

The working programme was reviewed and approved by the Academic Council of the Faculty of Information Technology

Chair of the Academic Council

 Tetiana HOVORUSHCHENKO

Khmelnyskyi 2025

2 LETTER OF APPROVAL

Position	Department Name	Signature	First Name, LAST NAME
Head of Department DSc, Prof.	Software Engineering		<u>Leonid BEDRATIUK</u>
Programme Guarantor DSc, Prof.	Software Engineering		<u>Leonid BEDRATIUK</u>

WEB TECHNOLOGIES

Type of Educational Component	Compulsory
Level of Higher Education	First (Bachelor's) Level
Language of Instruction	English
Semester	Fifth
Number of ECTS Credits Assigned	4
Forms of Study the Course is Designed For	Full-time

Learning Outcomes. *analyse*, purposefully search for and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology; *ability to choose* and use the appropriate software development methodology; *know and apply* in practice the fundamental concepts, paradigms and basic principles of functioning of language, tool and computing software engineering tools; *motivatedly choose* programming languages, etc.

Course Content. Basic concepts of web development. Addressing on the Internet. GIT version control systems. JavaScript technology. JavaScript: classes ES5 and ES6. Prototype inheritance. Functions of the constructor. ES6 classes. OOP in JavaScript: differences from others. Working with JSON, XML files. Using CMS. The concept of ASP.NET MVC technology. Working with controllers, models, views. Routing. Filters and working with them. Model binding. jQuery and AJAX. Authorisation and authentication in MVC.

Planned Learning Activities. The minimum amount of classroom-based learning activities in one ECTS credit for a course at the first (Bachelor's) level of higher education in full-time study mode is 10 hours per 1 ECTS credit.

Forms (Methods) of Instruction: Lectures (using problem-based learning and visualisation methods), Laboratory works, Independent work.

Assessment Methods: Laboratory work defence, Testing

Form of Final Assessment: Exam

Learning Resources:

1. WWW Consortium (W3C) Official Page. URL: <http://www.w3c.org/>.
2. Berners-Lee The founding of the web = Weaving the web. The original design and ultimate destiny of the world wide web: How the World Wide Web Began and Where It Will Go / Tim Berners-Lee with Mark Fischetti; translated from English by A. Ishchenko - Kyiv : Kyiv-Mohyla Academy, 2020. 208 p.
3. Tseslov O.V. WEB-programming: a textbook / O.V. Tseslov: Ministry of Education and Science, Youth and Sports of Ukraine, National Technical University of Ukraine "Kyiv Polytechnic Institute." – Kyiv : NTUU "KPI", 2020. 149 p.
4. Programming the Internet. Methodical instructions for the implementation of the course project by students of the speciality "Software Engineering". Print. Khmelnytskyi, KhNU, 2023. - 105 p.
5. Rajkumar Buyya, James Broberg, Andrzej Goscinski Cloud computing Principles and Paradigms. John Wiley & Sons, Inc., New Jersey, Canada, 2021. - 637 p.
6. Modular learning environment MOODLE. Access to the resource: <https://msn.khmnu.edu.ua/course/view.php?id=8661>.
7. Electronic library of the university. Access to the resource: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.

Lecturer: Candidate of Sciences, Associate Professor Yashyna O.M.

3. EXPLANATORY NOTE

The discipline "Web Technologies" is one of the professional disciplines and takes a leading place in the training of specialists of the bachelor's degree in the specialty 121 "Software Engineering" under the educational and professional programme "Software Engineering".

Prerequisites – CPT.04 Object-Oriented Programming, CPT.06 Databases.

Postquisites – CPT.01 Software Architecture and Design, CPT.12 Web Technologies (Course Project)

In accordance with the educational programme, the course contributes to the development of:

- **competences:** (IC) Ability to solve complex, specialised tasks or practical problems in software engineering, characterised by complexity and uncertainty of conditions, using information technology theories and methods.; (GC02) Ability to apply knowledge in practical situations; (PC08) Ability to apply fundamental and interdisciplinary knowledge to resolve software engineering tasks successfully.; (PC09) Ability to evaluate and consider economic, social, technological, and ecological factors affecting the professional activity field. (PC13) Ability to reasonably choose and master the toolkit for software development and maintenance.

- **software results training** : (PLO01) To analyse, purposefully search for, and select the necessary information, reference resources, and knowledge for solving professional tasks, considering modern scientific and technical achievements; (PLO06) To select and utilise a software development methodology appropriate for the task. (PLO07) To understand and apply in practice the fundamental concepts, paradigms, and basic principles of functioning linguistic, instrumental, and computational tools of software engineering; (PLO15) To make informed decisions when choosing programming languages and development technologies to address the tasks of creating and maintaining software.

- **The purpose of the course:** to form in students a system of knowledge, skills and abilities in the application of database design and development methods necessary for professional activity in the field of software engineering.

Subject of the course: The subject of the discipline is the principles of modern programming for the Internet space, as well as methods of their use in the development of web resources for various purposes.

Course objectives: students' mastering of basic knowledge of modern web technologies and basics of web programming, including web services technologies, ASP.NET MVC Web Framework, Entity Framework, programming using Javascript, jQuery, AJAX, mastering of theoretical and practical knowledge on the transformation, transmission and use of information in the global space, revealing the importance of information processes in the formation of a modern system and information scheme of the informatised world..

Learning outcomes: *analyse*, purposefully search for and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern advances in science and technology; *ability* to select and use a software development methodology appropriate to the task; *know* and apply in practice the fundamental concepts, paradigms and basic principles of functioning of language, tools and computing tools of software engineering; *motivatedly* choose programming languages and development technologies to solve problems of creating and maintaining software.

4. Structure of credits of the discipline “Web Technologies”

Topics	Lectures	Lab . work	Independent work of students
Topic 1. Methodologies and technologies of web Development.	6	8	18
Topic 2. JavaScript technology.	8	4	18
Topic 3. ASP.NET MVC technologies.	18	4	18
Together	32	34	54

5. COURSE PROGRAM “Databases”

5.1. Lecture course content *

Lecture No.	List of lecture topics, their abstract	Number of hours
Topic 1. Methodologies and technologies of web Development.		
1	Lecture 1. Basic concepts of web development. Functions and tasks of the developer. Web development methodologies. The concept of web applications and components for working with them. WWW service: components, software. Modern web browsers and principles of their work. Overview of CMS systems, frameworks, programming languages, web servers and server-side application programming technologies. Classical and agile web development methodologies: types, principles, rules. Information and reference resources for solving web development problems. Literature: [2, 5,7,8]	2
2	Lecture 2. Addressing in the Internet. Data transfer protocols. HTTP/HTTPS protocol. HTTP request methods. Client-server architecture. Transfer of user documents. References: [2, 5,7,8].	
3	Lecture 3. GIT version control systems. Basic concepts of version control systems. Installing and configuring GIT. Automation of workflows using GitHub Action. Literature: [7,8,9]	
Topic 2. JavaScript technology.		
4	Lecture 4. JavaScript basics. Code structure, data types and basic operators. Connecting JavaScript to a web page in different ways. Variables, data types and type conversions. Working with conditional statements JavaScript basics: objects, arrays, loops and functions. methods of objects and arrays. Map and Set collections, WeakMap, WeakSet. Work of loops and data processing for writing business logic of programs. Types of functions and possibilities of code reuse/ Literature: [2, 5, 7, 8].	2
5	Lecture 5. JavaScript: ES5 and ES6 classes. Prototype inheritance. Constructor functions. ES6 classes. OOP in JavaScript: differences from other languages. The 'this' keyword and call context. The main differences between JavaScript OOP and OOP in other languages. Splitting the code into independent modules Literature: [2,3,5,6,7,8].	
6	Lecture 6. Working with JSON and XML files. Working with different formats in the JavaScript programming language. Queries, event handling and user input. Setting up and sending AJAX requests. Creating DOM elements and adding them to the page. Properly requesting and retrieving data from the server. Handling user events Authorisation and authentication in browsers. Using regular expressions to check the correctness of user input. Different types of	

	authorisations. Cookie and session. Methods of storing data in the browser [2,3,5,6,7,8].	
7	Lecture 7. Using CMS. The history of the origin. CMS WordPress, Joomla and others. Differences between CMS and created sites on developed platforms [7,8,9].	
Topic 3. ASP.NET MVC technologies.		
8	Lecture 8. ASP.NET MVC technologies. The concept of ASP.NET MVC technology. Controller, Model, Model. View. Processing requests by MVC application. Approach to developing MVC applications. Features of ASP.NET MVC. Creating the first application. Literature: [2,3,5,6,7,8].	2
9	Lecture 9. Working with controllers. Basics of working with controllers. Methods of actions and their parameters. Results of actions. Forwarding and sending status and error codes. Sending files in ASP.NET MVC 5. The context of the request HttpContext. Cookies. Sessions. Asynchronous methods. Literature: [3,5,6,7,8].	2
10	Lecture 10. Working with representations. Basics of working with views. Strictly typed views. Master pages. Partial views. HTML helpers. Working with forms. Strictly typed helpers. Literature: [3,5,6,7,8].	2
11	Lecture 11. Working with models. Creating models. Connecting to the database Template helpers. Adding and deleting a model. Formation templates. Models with a complex structure and working with them. Models with many-to-many communication. Transfer arrays and complex data to the controller. Literature: [3,5,6,7,8].	2
12	Lecture 12. Working with models. Database migration. Creating a pagination. Overriding the formation templates. Overriding editing and image templates. Data filtering. Domain Model and View Model. Managing hierarchical data. Literature: [3,5,6,7,8].	2
13	Lecture 13. Routing. Defining routes. Working with routes. Creating restrictions for routes. Generating outgoing URL addresses. Areas. Create your own route handler. Routing attributes. Routing and nested resources. Literature: [3,5,6,7,8].	2
14	Lecture 14. Metadata and model validation. Data annotations for displaying properties. Basics of validation. Validation attributes. Validation of the model in the controller. Displaying validation errors. Creating your own validation logic. Literature: [3,5,6,7,8].	2
15	Lecture 16. Filters. Filters in ASP.NET MVC. Authentication filters. Authorisation filters. Model validation in the controller. Displaying validation errors. Creating your own validation logicLit. Literature: [3,5,6,7,8].	2
16	Lecture 17. Authorisation and authentication in MVC. Introduction to authorisation and authentication. Windows authentication. Authentication of forms. Authorisation in MVC and SimpleMembershipProvider. Setting up the use of SimpleMembershipProvider. Using universal providers. Web Site Administration Tool. Overriding the role provider. Profile provider and its overrides. OAuth - authentication. Literature: [3,5,6,7,8].	2
Together		32

5.2 Laboratory content

No. of the company	Laboratory topics	Hours
1	Principles of developing Web applications with using ASP.NET technology. Developing a simple MVC project in VS.	4

2	The basics of JavaScript.	4
3	Using ASP.NET MVC to create a web application. Developing a website for a bank or credit organisation.	4
4	ASP.NET content processing visualisation mechanism – Razor.	4
5	Models with a complex structure.	4
6	Validation of the model. Annotation of data for display of properties.	4
7	Basics of working with jQuery, jQuery UI and AJAX. JSON format.	4
8	Authorisation and authentication in MVC.	4
9	The final lesson.	2
Total for the semester		34

5.3 Content of independent work

The volume of independent work in the discipline "Web Technologies" is 54 hours, and 60 hours for course project . They include the study of lecture material, theoretical and laboratory tasks, preparation for laboratory work, their defense, and current testing.

Independent work of higher education applicants consists of systematic study of program material from relevant sources of information, preparation for laboratory classes, completion of individual tasks, testing of theoretical material, etc. In addition, students have access to the discipline page in the Modular Learning Environment, where the Discipline Work Program and necessary documents for its educational and methodological support are posted.

Week number	Topic name	Hours
1	Topic 1. Methodologies and technologies of web Development. Study of lecture material, performance of laboratory work No. 1. Literature: [2] p.20-50 [3] p.20-78	4
2	Topic 1. Methodologies and technologies of web Development. Study of lecture material. Defense of laboratory work No. 1. Literature: [2] p.20-50 [3] p.20-78	4
3	Topic 1. Methodologies and technologies of web Development. Study of lecture material, performance of laboratory work No. 2. Literature : Literature : [4] p.18-44 [6] p.35-59	4
4	Topic 1. Methodologies and technologies of web Development. Study of lecture material, performance of laboratory work No. 2. Literature : Literature : [4] p.18-44 [6] p.35-59	6
5	Topic 2. JavaScript technology . Study of lecture material. Defense of laboratory work No. 2. Literature: [4] p.18-44 [6] p.35-59	4
6	Topic 2. JavaScript technology. Study of lecture material. Performing laboratory work No. 3. Preparing for the test. Literature: [3] 450-480, [7] p.95-105, [10]	4
7	Topic 2. JavaScript technology. Study of lecture material. Defense of laboratory work No. 3. Literature: [7] p.95-105, [10].	4
8	Topic 2. JavaScript technology. Study of lecture material. Performance of laboratory work No. 4. Literature: [7] pp. 95-105, [10]. Testing.	6
9	Topic 3. ASP.NET MVC technologies. Study of lecture material. Defense of laboratory work No. 4. Literature: [3] p.401-412, [6] p.26-117. [10]	2
10	Topic 3. ASP.NET MVC technologies. Study of lecture material. Performance of laboratory work No. 5. Literature: [3] p.401-412, [6] p.26-117. [10]	2

12	Topic 3. ASP.NET MVC technologies. Study of lecture material. Defense of laboratory work No. 5. Literature: [3] p.595-602, [9] p.200-248 [1], p.618-648, [10]	2
13	Topic 3. ASP.NET MVC technologies. Study of lecture material. Performance of laboratory work No. 6. Literature: [3] p.595-602, [9] p.200-248 [1], p.618-648, [10]	2
14	Topic 3. ASP.NET MVC technologies. Study of lecture material, defense of laboratory work No. 6. Literature: [6] p.31-34, [10] [7] p.52-58, [9]	2
15	Topic 3. ASP.NET MVC technologies. Study of lecture material, performance of laboratory work No. 7. Literature: [6] p.31-34, [10] [7] p.52-58, [9]	4
16	Topic 3. ASP.NET MVC technologies. Study of lecture material, defense of laboratory work No. 7. Literature: [6] p.31-34, [10] [7] p.52-58, [9]	4
Total for the semester		54

6. TECHNOLOGIES AND TEACHING METHODS

The learning process for the course is based on the use of both traditional and modern teaching technologies and methods, in particular: lectures (using visualisation methods, problem-based and interactive learning, motivational techniques, and information and communication technologies); laboratory works (using training exercises, problem situation analysis, explanation, discussions, etc.); independent work (study of theoretical material, preparation for laboratory works, ongoing and final assessment), with the use of information and computer technologies and distance learning technologies.

7. METHODS OF ASSESSMENT

Ongoing assessment is carried out during practical classes, as well as on the days of control activities established by the working programme and the academic schedule.

The following methods of ongoing assessment are used:

- test-based assessment of theoretical material;
- evaluation of the results of laboratory work defence.

When determining the final semester grade, the results of both ongoing assessment and final assessment are taken into account. The final assessment is conducted on all the material of the course according to examination papers prepared in advance and approved at the meeting of the department.

A student who has scored less than 60 percent of the maximum score for any type of academic work is not allowed to undergo the semester assessment until the amount of work stipulated by the Working Programme is completed. A student who has achieved a positive weighted average score (60 percent or more of the maximum score) for all types of ongoing assessment but has failed the examination is considered to have an academic debt.

Elimination of academic debt for the semester assessment is carried out during the examination session or according to the schedule set by the dean's office in accordance with the *Regulation on Control and Assessment of Learning Outcomes of Students at Khmelnytskyi National University*.

8. COURSE POLICY

The policy of the academic course is generally determined by the system of requirements for the student as stipulated by the current University regulations on the organisation and teaching and learning support of the educational process. In particular, this includes completing the safety briefing; attendance at course classes is compulsory. For valid reasons (documentarily confirmed), theoretical training may, with the lecturer's approval, take place online. Successful completion of the course and the formation of professional competences and programme learning outcomes require preparation for each laboratory work (studying the theoretical material for the topic of the work), active participation during the class, thorough preparation of the report, defence of the results, participation in discussions regarding the constructive decisions made during the laboratory works, etc.

Students must meet the established deadlines for completing all types of academic work in accordance with the Working Programme of the course. A missed laboratory class must be completed within the deadline set by the lecturer, but no later than two weeks before the end of the theoretical classes in the semester.

The student's mastery of the theoretical material of the course is assessed through testing.

When performing laboratory work, the student must comply with the policy of academic integrity (cheating, plagiarism — including with the use of mobile devices — is prohibited). If a violation of academic integrity is detected in any type of academic work, the student receives an unsatisfactory grade and must re-do the task on the relevant topic (type of work) as stipulated by the Working Programme. Any form of academic dishonesty is unacceptable.

Within the framework of studying the course, students are provided with recognition and crediting of learning outcomes acquired through non-formal education, available on accessible platforms (<https://prometheus.org.ua/>, <https://www.coursera.org/>), which contribute to the formation of competences and the deepening of learning outcomes defined in the Working Programme of the course, or ensure the study of a relevant topic and/or type of work from the course syllabus (for more details, see the *Regulation on the Procedure for Recognition and Crediting of Learning Outcomes of Students at Khmelnytskyi National University*).

9. ASSESSMENT OF STUDENTS' LEARNING OUTCOMES DURING THE SEMESTER

Assessment of a student's academic achievements is carried out in accordance with the *Regulation on the Control and Assessment of Students' Learning Outcomes at Khmelnytskyi National University*. During the ongoing assessment of the work performed by the student for each structural unit and the results obtained, the lecturer awards a certain number of points as set out in the Working Programme for that type of work.

Each structural unit of academic work may be credited only if the student has scored at least 60 percent (the minimum level for a positive grade) of the maximum possible points assigned to that structural unit.

When assessing students' learning outcomes for any type of academic work (structural unit), it is recommended to use the generalised criteria provided below:

Table – Assessment Criteria for Student Learning Outcomes

Grade and Level of Achievement of Intended Learning Outcomes and Competences	General Description of Assessment Criteria
Excellent (High)	The student has deeply and fully mastered the course content, confidently navigates it, and skilfully uses the conceptual framework; demonstrates the ability to connect theory with practice, solve practical problems, and clearly express and justify their reasoning. An excellent grade implies a logical presentation of the answer in the language of instruction (oral or written), high-quality formatting of the work, and proficiency in using specialised tools, instruments, or application software. The student demonstrates confidence when answering reformulated questions, is capable of making detailed and summarised conclusions, and shows practical skills in solving professional tasks. The answer may contain two or three minor inaccuracies.

Grade and Level of Achievement of Intended Learning Outcomes and Competences	General Description of Assessment Criteria
Good (Average)	The student has shown full understanding of the course content, possesses the conceptual framework, and navigates the material well; applies theoretical knowledge consciously to solve practical tasks. The answer is generally well-articulated, although some minor inaccuracies or vague formulations of rules or principles may occur. The student's answer is based on independent thinking. Two or three minor mistakes are acceptable.
Satisfactory (Sufficient)	The student demonstrates knowledge of the basic course material sufficient for continued learning and practical activity in the profession; is able to complete the practical tasks foreseen by the programme. The answer is usually based on reproductive thinking. The student has limited knowledge of the structure of the discipline, makes inaccuracies and significant errors in the answer, and hesitates when answering reformulated questions. Nevertheless, they possess basic skills to complete simple practical tasks that meet the minimum assessment criteria and, under the lecturer's guidance, can correct their mistakes.
Unsatisfactory (Insufficient)	The student demonstrates fragmented, unstructured knowledge, cannot distinguish between main and secondary ideas, makes conceptual errors, misinterprets definitions, presents material in a chaotic and unconfident manner, and cannot apply knowledge to solve practical problems. An unsatisfactory grade is typically given to a student who is unable to continue learning the subject without additional study.

Structuring of the Course by Types of Academic Work and Assessment of Student Learning Outcomes

Classroom work								Control measures	Semester control	Together
<i>Second semester</i>										
Laboratory work No.:								Test control:	Exam	Total points
1	2	3	4	5	6	7	8	T 1		
Number of points per type of academic work (minimum-maximum)										
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	12-20		
24-40								12-20	24-40	60-100

Symbols : T – subject topic

Notes: If the number of points earned for any type of academic work in the course is below the established minimum, the student receives a failing grade and must retake the work within the deadline set by the lecturer (or dean). The institutional grade is determined in accordance with the table "**Correspondence between the Institutional Grading Scale and the ECTS Grading Scale**".

Assessment of Laboratory Work Defence Results

A laboratory work completed and formatted in accordance with the requirements established in the Methodological Guidelines is comprehensively assessed by the lecturer during its defence based on the following criteria:

- independence and accuracy of execution;
- completeness of the answer and understanding of the principles of building machine learning models;
- ability to justify the choice of algorithm or method;
- correctness of model implementation in the Python programming environment using appropriate libraries;
- ability to interpret the results of modelling and evaluate their suitability for solving the given task.

When assessing a laboratory session, the lecturer uses the generalised criteria outlined in the table “Assessment Criteria for Student Learning Outcomes” (minimum passing score – 3 points, maximum – 5 points).

If the student demonstrates a knowledge level below 60 percent of the maximum score established in the Working Programme for each structural unit, the laboratory work is not credited. In such a case, the student must study the topic more thoroughly, review the methodology, correct major mistakes, and re-defend the work at the time set by the lecturer.

Assessment of Test-Based Control Results

Each test included in the Working Programme consists of 30 test items, each carrying equal weight. According to the table for structuring types of academic work, the student may receive between 3 and 5 points depending on the number of correct answers.

Distribution of points depending on the number of correct answers to test items:
The test duration is 30 minutes. Students complete the test online in the Modular Learning Environment.

If a failing grade is received, the test must be retaken before the next scheduled assessment.

Distribution of points depending on correct answers to test questions

Number of Correct Answers	1-17	18-23	24-26	27-30
Percentage of Correct Answers	0-59	60-79	80-89	90-100
Number of Points	-	3	4	5

The test is given 30 minutes. The student takes the test online in the Modular Learning Environment. If the test is negative, the test must be retaken before the **next** control deadline.

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is provided in the table “Correspondence” below.

Assessment of the Final Semester Control (Exam)

The educational programme provides for a final semester control in the form of an examination, the purpose of which is to systematically and objectively assess both the theoretical and practical preparation of the student in the course. The examination is conducted according to examination papers prepared in advance and approved at the meeting of the department. In accordance with this, the examination paper contains a combination of both theoretical questions (including in test form) and practical tasks.

Table – Assessment of Final Semester Examination Results for full-time students (40 points allocated for final control)

Type of Task	For each individual type of task		
	Minimum (Satisfactory) Score	Potential Positive Score (Good)*	Maximum (Excellent) Score
Theoretical Question № 1	3	4	5
Theoretical Question № 2	3	4	5
Practical Tasks (6 tasks worth 3 points each)	18	24	30
Total:	24		40

Note. A passing score for the exam, different from the minimum (24 points) and the maximum (40 points), falls within the range of 25–39 points and is calculated as the sum of points for all structural elements (tasks) of the exam.

For each individual type of task in the final semester assessment, the assessment criteria for student learning outcomes provided above (see **Table – Assessment Criteria for Student Learning Outcomes**) are applied.

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is shown below in the **Correspondence Table**.

The final examination grade is recorded if the total number of points accumulated by the student in the course as a result of ongoing assessment falls within the range of 60 to 100 points. In this case, a grade of *Excellent/Good/Satisfactory* is assigned according to the institutional scale, and a letter grade is assigned according to the ECTS scale, corresponding to the total number of points earned by the student as specified in the **Correspondence Table**.

Table – Correspondence between the Institutional Grading Scale and the ECTS Grading Scale

ECTS Grade	Rating Scale (Points)	Institutional Grade(Level of Achievement of the Intended Learning Outcomes in the Course)	
		Pass/ Fail	Exam / Graded Credit
A	90-100	Pass	Excellent – a high level of achievement of the intended learning outcomes in the course, indicating the learner’s full readiness for further study and/or professional activity in the field.
B	83-89		Good – an average (maximally sufficient) level of achievement of the intended learning outcomes in the course and readiness for further study and/or professional activity in the field.
C	73-82		
D	66-72		Satisfactory – the student has demonstrated a minimally sufficient level of achievement of the learning outcomes required for further study and/or professional activity in the field.
E	60-65		

FX	40-59	Fail	Fail – several intended learning outcomes in the course have not been achieved. The level of acquired learning outcomes is insufficient for further study and/or professional activity in the field.
F	0-39		Fail – no learning outcomes have been achieved.

10. Questions for student test control from the discipline "Web Technologies"

1. What is a web application?
2. What is a browser?
3. Describe the cycle of processing a request to a web application from a client.
4. Why do you need web application development technologies (such as ASP.NET, PHP, Ruby On Rails, etc.)?
5. How does the HTTP protocol work and what is it for?
6. What are HTTP message headers and what are they for?
7. What is the body of an HTTP message?
8. How are headers separated from the message body in an HTTP message?
9. What is the HTTP request method?
10. What is the HTTP response status code?
11. Give examples of HTTP headers for an HTTP request and an HTTP response.
12. How does the secure HTTPS protocol work?
13. What is a web server?
14. What interfaces can a web server and a web application use to communicate?
15. What is the difference between CGI and ISAPI?
16. What is shared hosting?
17. What is an application pool?
18. Name the most popular implementations of web servers.
19. What kind of web server do ASP.NET applications run on?
20. Name the stages of MVC application development.
21. Describe the request lifecycle for an MVC application.
22. Microsoft ASP.NET MVC is a?
23. What is a Controller?
24. What is the Model?
25. What is a Submission?
26. The algorithm of the MVC application.
27. What tasks is the ASP.NET MVC platform designed to solve?
28. The extension methods are LINQ.
29. The from construct and range variables are LINQ.
30. The join construction is LINQ.
31. The orderby construction is LINQ.
32. The select construction and projection are LINQ.
33. The let construct is LINQ.
34. The group construction is LINQ.
35. The in and continuation constructs are LINQ.
36. The basics of controllers.
37. Defining the IController interface.
38. Methods of action.

39. Receiving input data. Transferring parameters.
40. Classes derived from ActionResult. 60. Passing data from the controller to the view.
41. Redirection. Objects RedirectResult, RedirectPermanent.
42. The RedirectToAction method.
43. The ContentResult.
44. File transfer. HTTPCONTEXT.
45. The context of the request.
46. Cookies.
47. Introduction to the view.
48. Representation using Razor syntax.
49. Paths to view files.
50. Types of presentation templates.
51. Strictly typed representations.
52. Razor view engine.
53. Master pages.
54. The _ViewStart.
55. Partial representations.
56. Helper Html.Partial. 77. Helper Html.RenderPartial.
57. HTML headers.
58. Strictly typed helpers. Lambda expressions.
59. Models and databases.
60. Entity Framework.
61. Connecting to the database.
62. Template helpers.
63. Edit the model.
64. Form templates.
65. Models with a complex structure.
66. Working with complex models.
67. Data transfer to the controller.
68. Routing.
69. Defining routes.
70. Working with routes.
71. Create restrictions for routes.
72. Generate outgoing URLs.
73. Areas.
74. Create your own route handler.
75. Metadata.
76. Model validation.
77. Data annotations for displaying properties.
78. Validation attributes.
79. Validation of the model in the controller.
80. Display of validation errors.
81. Create your own validation logic.
82. Filters.
83. Authorisation filters.
84. Exception filters.
85. Filters for actions and results.
86. Global filters.
87. Built-in filters.
88. Linking the model.
89. Create a model binder.

90. jQuery and AJAX.
91. Using JavaScript/jQuery.
92. Ajax Forms.
93. Parameters of the AjaxOptions object.
94. AJAX link.
95. JSON format.
96. AJAX requests using jQuery.
97. Working with jQuery UI.
98. Authorisation and authentication in MVC.

11. Questions for the final test of students in the discipline "Web Technologies"

1. What symbol marks the beginning of C# code insertions in Razor?
2. At what level is the HTTP protocol located?
3. The InnerHtml property allows you to set or get the content of a tag as a string.
4. The BeginForm method takes the name of the action method and the name of the controller as parameters, as well as the type of request.

12. Educational and methodological software

Educational process in the discipline "Web Technologies" completely and in sufficient quantities provided necessary educational and methodological literature. In particular, teachers' departments prepared and published such works:

-Web technologies: methodological recommendations for laboratory work for applicants of the first (bachelor) level of higher education, specialty 121 "Software engineering" = Вебтехнології : методичні рекомендації до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення» / О. М. Яшина, В. В. Мартинюк. Хмельницький : ХНУ, 2024. 102 с. (англ., укр.)

13. Hardware and software software disciplines (if necessary)

Information and computer support: PC, tablet, smartphone or another mobile device, projector.
Software software: programs Microsoft Office or similar , network access Internet, working with presentations.

Study educational doesn't need discipline using special software application software, them commonly used programs and operating systems.

14. Recommended reading

Basic:

1. Programming the Internet. Methodical instructions for the implementation of the course project by students of the speciality "Software Engineering". Print. Khmelnytskyi, KhNU, 2023. 55 p.
2. WWW Consortium (W3C) Official Page. (<http://www.w3c.org/>).
3. Berners-Lee The founding of the web = Weaving the web. The original design and ultimate destiny of the world wide web: How the World Wide Web Began and Where It Will Go / Tim Berners-Lee with Mark Fischetti; translated from English by A. Ishchenko - Kyiv : Kyiv-Mohyla Academy, 2018. 208
4. Web programming: a textbook / O.V. Tseslov ; Ministry of Education, Youth and Sports of Ukraine, National Technical University of Ukraine "Kyiv Polytechnic Institute." - Kyiv : NTUU "KPI", 2020. 149 p.
5. Schmidt, J. The New Network: Signs, Practices and Consequences of Web 2.0 = Das Neue Nets Markmale, Praktiken und Folgen des Web 2.0 : a textbook for universities / Jan Schmidt ; [translated from German by V. Klymchenko ; edited by V. Ivanov] - Kyiv : Academy of Ukrainian

- Press, Free Press Centre, 2013. 283 p.
6. What is front-end development. URL: <https://www.freecodecamp.org/news/front-end-developer-what-is-front-end-development-explained-in-plain-english/>.
 7. Front-end development technologies concepts. URL: <https://www.altexsoft.com/blog/front-end-development-technologies-concepts/>.
 8. Beginners guide for back-end development. URL: <https://www.upwork.com/resources/beginners-guide-back-end-development>.
 9. Back-end architecture. URL: <https://www.codecademy.com/article/backend-architecture>.
 10. Development tools for web developers. URL: <https://www.geeksforgeeks.org/12-backend-development-tools-for-web-developers/>.
 11. What back-end does your website need? URL: <https://goldwebsolutions.com/uk/blog/yakij-backend-potribnij-vashomu-vebsajtu/>.

Support

12. Oliver Dick. Popular Web browsers. User's encyclopedia. - K.: DiaSoft, 2018. - 280 p.
13. Bowen Rich. Apache. Administrator's Handbook - K.: DiaSoft, 2019. - 350 c.
14. Matvienko, O.V. Internet-technologies: designing a web page: A textbook for university students / O.V. Matvienko, I.L. Borodkina. - 2nd edition, revised and supplemented. - K.: TSNL, 2019. - 154 c.
15. Kulenko M.Y. Fundamentals of graphic design: a textbook for students of higher educational institutions / Mykhailo Kulenko; MESU; Kyiv National University of Construction and Architecture. 2nd edition, revised and supplemented - Kyiv : Condor, 2020. 492 p.

15. Information resources

Electronic University:

1. A modular learning environment. Access to the resource: <https://msn.khmnu.edu.ua/course/view.php?id=8661>
2. Electronic library of the university. Access to the resource: <https://lib.khnu.km.ua>
3. Repository of KhNU. Access to the resource: <https://elar.khmnu.edu.ua/home>