General information	**Name, code and number of credits**	CMS220, Web Programming-2, 3 credits, 6 ECTS
	Departament	Computer Science
	Program (bachelors, master)	Bachelor
	Semester	Spring 2026
	Subject teacher(s)	Leyla Aliyeva
	E-mail:	leylaa.aliyeva1@gmail.com
	Lecture room	401n
	Advice hours	
Prerequisites	Basic Computer Skills	
Language of instruction	English	
Type of subject (compulsory/elective)	Major	
Resources	Textbooks: 1. Node.js Design Patterns, Mario Casciaro, Luciano Mammino, 3rd Edition, Packt Publishing, 2020 2. Eloquent JavaScript, Marijn Haverbeke, 3rd Edition, No Starch Press, 2018 (Online Edition Available) 3. Web Development and Design Foundations with HTML5 Terry Felke-Morris, 9th Edition, Pearson, 2021 Additional Resource Texts: 4. Learning Node.js Development, Shelley Powers, Alex Young, Marc Harter, Addison-Wesley, 2017. 5. Express in Action, Evan M. Hahn, Manning Publications, 2016 6. You Don't Know JS Yet, Kyle Simpson, O'Reilly Media, 2020 Auxiliary Web sources: 1. MDN Web Docs: https://developer.mozilla.org/en/ 2. W3Schools: https://www.w3schools.com/ 3. Express Js Docs: https://expressjs.com/ 4. Node Js Docs: https://nodejs.org/en/docs/guides/getting-started-guide/	
Course description	This course focuses on server-side web programming and the development of dynamic, database-driven web applications using Node.js and Express.js. Students will learn how web servers handle HTTP requests and responses, process user input, manage sessions and cookies, and interact with relational databases using SQL. The course emphasizes practical implementation of backend functionality, including REST-style CRUD operations, user authentication and authorization, input validation, and basic security practices. Through hands-on laboratory work and a semester-long project, students will design, implement, and deploy a complete server-side web application. By the end of the course, students will gain foundational skills in backend	

<table>
<tr><td></td><td colspan="3">development and an understanding of how server-side components integrate with frontend applications to form full web systems.</td></tr>
<tr><td>Course objectices</td><td colspan="3">The objectives of this course are to:
— Introduce fundamental concepts of server-side web programming and the client–server architecture.
— Explain how HTTP requests and responses are handled on the server side.
— Develop practical skills in building RESTful web services that process user input and generate dynamic content.
— Teach the basics of database-driven web applications using relational (SQL) databases.
— Provide hands-on experience with session management, cookies, and user authentication.
— Introduce essential web security practices, including input validation and secure password handling.
— Enable students to design, implement, and present a complete server-side web application using modern tools and frameworks.</td></tr>
<tr><td>Learning outcomes</td><td colspan="3">By the end of this course, students will be able to:
— Explain the principles of client–server communication and the HTTP request–response lifecycle.
— Develop server-side applications that handle GET and POST requests.
— Implement CRUD (Create, Read, Update, Delete) operations using RESTful API design.
— Process and validate user input received from web forms.
— Design and interact with relational databases using basic SQL queries.
— Manage user sessions and cookies to maintain application state.
— Implement user registration, login, and role-based access control.
— Apply basic web security techniques such as input validation and password hashing.
— Organize server-side code following simple structural and modular principles.
— Deploy a basic server-side web application and present its functionality.</td></tr>
<tr><td rowspan="4">Teaching methods</td><td>Lecture</td><td colspan="2">+</td></tr>
<tr><td>Group discussions</td><td colspan="2">+</td></tr>
<tr><td>Experiential exercise</td><td colspan="2">+</td></tr>
<tr><td>Course paper</td><td colspan="2"></td></tr>
<tr><td rowspan="4">Assessment</td><td>Components</td><td>Date/deadline</td><td>Percentage (%)</td></tr>
<tr><td>Active participation</td><td>At each lesson</td><td>10</td></tr>
<tr><td>Homework Assignments</td><td>At each lesson</td><td>10</td></tr>
<tr><td>Attendance</td><td>At each lesson</td><td>10</td></tr>
</table>

	Midterm exam		30
	Final exam		40
	Final		**100**

Rules (Teaching policy and behaviour)

Preparation for Class:

Students are expected to complete all assigned readings and exercises prior to class to ensure active participation during lectures and lab sessions.

1. Withdrawal (pass/fail)

- This course strictly follows the grading policy of the School of Engineering and Applied Science. Thus, a student is normally expected to achieve a mark of at least 60% to pass. In case of failure, he/she will be required to repeat the course the following term or year.

2. Cheating/Plagiarism:

- Strictly prohibited. Any form of cheating or plagiarism will result in a zero for the assignment/exam and may lead to further academic consequences.

3. Ethics

- Students should not arrive late to class.
- All cell phones must be turned off and stowed away before entering class.
- Use of any electronic devices is not allowed in the classroom and violators will be punished accordingly.

4. Active Participation (10%)

- Students are expected to contribute actively during each lesson.
- Participation includes answering questions, engaging in discussions and taking part in group activities.
- Disruptive or passive behavior will negatively affect participation grades.

5. Homework Assignments (10%)

- Homework is assigned at each lesson and must be submitted on time.
- Late submissions will not be accepted unless there is a valid and documented reason.

6. Attendance (10%)

- Attendance is mandatory for all lessons.
- Students who miss more than the allowed number of classes (without a valid excuse) will lose attendance points.
- Consistent lateness may also reduce attendance grades.

7. Midterm exam (30%)

- The midterm exam will cover material from the first half of the course.
- The exam may include multiple-choice questions, short answers and

	problem-solving tasks. 8. Final exam (40%) ● The final exam will cover the entire course content with greater emphasis on material taught after the midterm. ● The format will be similar to the midterm exam. ● Students must take the final exam to pass the course.

Tentative schedule

Week	Date	Topics to be covered	Tasks/Resources
1.	16-02-2026	*Introduction to Server-Side Programming, Node.js and Express* **Learning Focus:** Client–server architecture and backend responsibilities HTTP request–response overview Node.js runtime environment Express.js framework basics Static and dynamic server responses **Applied Activities:** Installing Node.js Creating a basic Node.js script Building a simple Express server	[1] Chapter 1 – Introduction to Node.js and backend concepts [2] Chapter 11 – Asynchronous programming overview Additional reading https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works https://nodejs.org/en/docs/guides/getting-started-guide/
2.	23-02-2026	*HTTP Methods and CRUD Operations* **Learning Focus:** Purpose of HTTP methods GET, POST, PUT and DELETE How clients send data to servers Data exchange using JSON format CRUD lifecycle in web applications **Applied Activities:** Handling GET and POST requests Implementing basic CRUD endpoints Testing requests using Postman	[1] Chapter 2 – Modules and application structure Additional reading https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods https://developer.mozilla.org/en-US/docs/Glossary/CRUD
3.	02-03-2026	*Form Processing and Input Validation* **Learning Focus:** Form submission workflow Request body parsing Importance of server-side validation	[3] Chapter 9 – Forms and data handling Additional reading https://developer.mozilla.o

		User-friendly error handling **Applied Activities:** Processing form data Validating required fields Returning validation error messages	rg/en-US/docs/Learn/Forms https://expressjs.com/en/guide/routing.html
4.	09-03-2026	*Sessions and Cookies* **Learning Focus:** Stateless nature of HTTP Session management concepts Cookies and client-side storage Basic security considerations **Applied Activities:** Creating session-based login state Setting and reading cookies	[1] Chapter 6 – State and session-related concepts Additional reading https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/sessions
5.	16-03-2026	*File Upload and Static Content* **Learning Focus:** File upload workflow Handling multipart form data Serving static files securely **Applied Activities:** Uploading files from forms Accessing uploaded files via URL	[1] Chapter 4 – Input/output handling concepts Additional reading https://developer.mozilla.org/en-US/docs/Web/API/File https://expressjs.com/en/resources/middleware/multer.html
6.	23-03-2026	*Relational Databases and SQL Basics* **Learning Focus:** Relational database concepts Tables, rows and columns Primary and foreign keys Basic SQL queries **Applied Activities:** Creating database tables Inserting and retrieving records	[3] Chapter 11 – Databases and data storage concepts Additional reading https://developer.mozilla.org/en-US/docs/Learn/Server-side/Databases https://www.w3schools.com/sql/
7	30-03-2026	*Database Relationships and Backend Integration*	[3] Chapter 11 – Database relationships and queries

		Learning Focus: One-to-many database relationships Connecting backend to database Querying relational data **Applied Activities:** Linking tables using foreign keys Fetching related data	Additional reading https://www.w3schools.com/sql/sql_join.asp
8	06-04-2026	***Midterm Exam*** **Learning Focus:** Review of all topics covered so far Evaluation of backend fundamentals Applied Activities: Revision of Node.js, Express, HTTP, CRUD and SQL	Resources Revision of Weeks 1–7 Practice previous labs and examples
9	13-04-2026	*User Authentication* **Learning Focus:** User registration workflow Login process Authentication logic Password handling principles **Applied Activities:** Creating registration endpoint Implementing login functionality	[1] Chapter 8 – Authentication patterns Additional reading https://developer.mozilla.org/en-US/docs/Learn/Server-side/Authentication https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
10	20-04-2026	*Authorization and Protected Routes* **Learning Focus:** Difference between authentication and authorization User roles and permissions Middleware concept in Express **Applied Activities:** Protecting routes using middleware Implementing role-based access	[1] Chapter 8 – Authorization and access control Additional reading: https://expressjs.com/en/guide/using-middleware.html https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication
11	27-04-2026	*Code Organization and Environment Configuration* **Learning Focus:** Backend project structure	[1] Chapter 5 – Application structure and configuration Additional reading:

		Separation of concerns Environment variables and configuration **Applied Activities:** Refactoring project folders Using environment variables	https://expressjs.com/en/advanced/best-practice-performance.html https://www.npmjs.com/package/dotenv
12	04-05-2026	*API Design Best Practices* **Learning Focus:** Consistent API responses Success and error response formats Pagination basics **Applied Activities:** Standardizing API responses Implementing pagination	[1] Chapter 10 – API design principles Additional reading: https://developer.mozilla.org/en-US/docs/Glossary/REST https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
13	11-05-2026	*Error Handling and Security Basics* **Learning Focus:** Global error handling Common backend errors Basic security practices such as validation and rate limiting **Applied Activities:** Creating centralized error handler Preventing invalid requests	[1] Chapter 9 – Error handling strategies Additional reading: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors https://expressjs.com/en/guide/error-handling.html
14	18-05-2026	*Deployment Fundamentals* **Learning Focus:** Development versus production environments Preparing backend for deployment Deployment workflow overview **Applied Activities:** Configuring environment variables for production Running application in production mode	[1] Chapter 11 – Deployment considerations Additional reading: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/deployment https://12factor.net/
15	25-05-2026	*Final Project and Presentation* **Learning Focus:** Integrating all backend components Testing and debugging Presenting system design and functionality	Resources Review all course materials Project documentation and lecture notes

		Applied Activities: Final project completion Code cleanup and testing	
Final exam			