Syllabus
Course Program

NodeJS-based web applications

Specialty
121 – Software Engineering
122 – Computer Science

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering
Computer Science and Intelligent Systems

Department
Software Engineering and Management Intelligent Technologies (321)

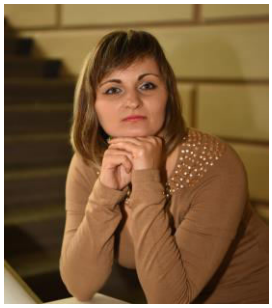
Level of education
Bachelor's level

Course type
Elective

Semester
6

Language of instruction
English, Ukrainian

Lecturers and course developers



Karina Melnyk

Karina.Melnyk@khpi.edu.ua
Ph.D., Associate Professor, Associate Professor of Software Engineering and Information Technology Management

Author (co-author) of more than 70 publications, 5 collective monographs, 10 articles in publications indexed in Scopus and Web of Science. (h-index = 5, i10-index = 1 in Google Scholar -https: //scholar.google.com/citations? user = xCU7GMgAAAAJ & hl = ru; ORCID ID https://orcid.org/0000-0001- 9642-5414; Scopus Author ID https://www.scopus.com/authid/detail.uri?authorId=57195074119).
Leading lecturer of the courses: Basics of Software Engineering (Bachelors) (in English), Methods of Empirical Information Processing (Bachelors) (in English and Ukrainian), Basics of Intelligent Systems Design (Masters) (in English and Ukrainian)
More about the lecturer on the department's website

General information

Summary

The discipline "NodeJS-based web applications" is an educational discipline of free choice of a student of specialized training in the specialties 121 "Software engineering", 122 "Computer science". The educational discipline is aimed at forming a complete picture of knowledge on creating network applications in the Javascript language, which can be launched outside the client's browser.

Course objectives and goals

The purpose of studying the academic discipline is to provide specialists with theoretical knowledge and practical skills in designing and creating server applications in the Javascript language.

Format of classes

Lectures, laboratory classes, consultations, self-study. Final control in the form of an exam.

Competencies

121 – Software engineering:

K13. Ability to identify, categorize and formulate software requirements.
K14. Ability to participate in the design of software, including modeling (formal description) of its structure, behavior and functioning processes.
K15. Ability to develop architectures, modules and components of software systems.
K19. Knowledge of data information models, ability to create software for data storage, extraction and processing.
K25. The ability to reasonably choose and master software development and maintenance tools.

122 – Computer science

PC8. Ability to design and develop software using various programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, calculation methods and algorithms, data structures and control mechanisms.
PC9. The ability to implement a multi-level computing model based on client-server architecture, including databases, knowledge and data warehouses, to perform distributed processing of large data sets on clusters of standard servers to meet the computing needs of users, including on cloud services.
PC13. Ability to develop network software that functions on the basis of various topologies of structured cabling systems, uses computer systems and data transmission networks, and analyzes the quality of computer networks.
PC20. The ability to develop the architecture of software systems and their individual components when building intelligent control systems in various industries, to manage the life cycle processes of the software of intelligent control systems.

Learning outcomes

121 – Software engineering:

PR12. Apply effective software design approaches in practice.
PR14. Apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and documentation of software.
PR15. Motivated to choose programming languages and development technologies to solve the tasks of creating and maintaining software.
PR18. Know and be able to apply information technologies for data processing, storage and transmission.

122 – Computer science

PR9. To develop software models of subject environments, to choose a programming paradigm from the standpoint of convenience and quality of application for the implementation of methods and algorithms for solving problems in the field of computer science.
PR10. Use tools for the development of client-server applications, design conceptual, logical and physical models of databases, develop and optimize queries to them, create distributed databases, data stores and showcases, knowledge bases, including on cloud services, using web languages -programming.
PR20. Develop the architecture of software systems and their separate components when building intelligent control systems in various industries, as well as manage the life cycle processes of the software of intelligent control systems.

Student workload

The total volume of the discipline is 150 hours (5 ECTS credits), in particular: lectures - 32 hours, laboratory hours - 32 hours, independent work - 86 hours. It is taught in the sixth semester. The discipline ends with a credit.

Course prerequisites

The course "NodeJS-based web applications" is based on the disciplines "JavaScript frameworks", "Advanced web development course".

Features of the course, teaching and learning methods, and technologies

Teaching and learning methods:
interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning.

Assessment forms:
written individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), online tests (CAS), final/semester control in the form of a semester credit, according to the schedule of the educational process (FAS) .

Program of the course

Topics of the lectures

Topic 1. Backend development.
What is backend development. Introducing backend JavaScript
Topic 2. Basics of Node.js.
Why you should use Node.js. How Does Node.js Work. Event loop and reactor pattern. How to Install Node.js
Topic 3. Program Structure in Node.js
Node.js Modules. Node.js NPM.
Topic 4. Advanced topics
Streams. EventEmitter. Exception handling.
Topic 5. Design Patterns. Best Practices.
Observer. Callbacks. Events. Code Organisation.
Topic 6 Node.js MySQL
Node.js MySQL Create Database. Node.js MySQL CRUD Tables.
Topic 7 Node.js MongoDB
Node.js MongoDB Create Database. Node.js MongoDB CRUD Collections.
Topic 8. Node.js Applications
Real-time chatting. Data streaming. System monitoring dashboards. REST API.

Topics of the workshops

Practical classes within the discipline are not provided.

Topics of the laboratory classes

Topic 1. Building a simple Node.js application.
Topic 2. Creating a Node.js + MySQL/PostgreSQL web application.
Topic 3. Creating a Node.js + MongoDB web application.
Topic 4. Creating a web application on the chosen topic.

Self-study

Students are recommended additional materials (videos, articles) for independent study and processing.

Course materials and recommended reading

1. Lim, Greg. Beginning Node.js, Express & MongoDB Development. Сінґапур, Greg Lim, 2020.
2. Herron D. Node.js Web Development. Third ed. Birmingham: Packt Publishing; 2016. https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=4594292.

3. Nandaa A. Beginning Api Development with Node. Js : Build Highly Scalable Developer-Friendly Apis for the Modern Web with Javascript and Node. Js. Birmingham: Packt Publishing; 2018. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1856582.
4. Hibbard J Reyes C Wanyoike M et al. Node. Js. Sebastopol: SitePoint Pty Limited; 2018. https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5611119
5. Rappl F. Modern Frontend Development with Node. Js a Compendium for Modern Javascript Web Development Within the Node.js Ecosystem. 1st ed. Birmingham: Packt Publishing Limited; 2022. https://public.ebookcentral.proquest.com/choice/PublicFullRecord.aspx?p=30254093.

Assessment and grading

Criteria for assessment of student performance, and the final score structure

100% final assessment in the form of credit (10%) and current assessment (90%).
10% credit: semester credit, according to the schedule of the educational process
90% current assessment:
70% assessment of tasks in laboratory works;
20% intermediate control (2 independent individual works)

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by 08.06.2023 Head of the department
Ihor HAMAIUN

08.06.2023 Guarantors of the educational programs
Andrii KOPP
Uliya LITVINOVA