Syllabus
Course Program

Server application programming

Specialty
172 – Electronic communications and radio engineering

Institute
Institute of Computer Modeling, Applied Physics and Mathematics

Educational program
Network technologies and telecommunications

Department
Information systems named after V.O. Kravets (169)

Level of education
Master's level

Course type
Elective

Semester
1

Language of instruction
English

Lecturers and course developers

Maksym Tolkachov
maksym.tolkachov@khpi.edu.ua

Associate professor of NTU "KhPI", associate professor of the department "Information systems named after V. O. Kravets " NTU "KhPI". Experience - 23 years. Author over 30 scientific and educational and methodical Works. Leading lecturer in the disciplines : "Computer and telecommunication networks", "Design and administration of computer networks", "Technologies of transport networks", "System software of information communication systems
"More about the lecturer on the department's website

General information

Summary

This course provides a comprehensive introduction to server-side application programming, focusing on building robust, scalable, and secure server applications for telecommunication systems. Students will explore key concepts such as server architecture, API development, database integration, and real-time communication. Practical projects and hands-on exercises will enable learners to design, implement, and optimize server applications for modern networked environments..

Course objectives and goals

The primary goal of this course is to equip students with the skills and knowledge required to develop high-performance server applications. By covering essential programming principles, frameworks, and tools, students will learn to create reliable back-end systems that efficiently handle data, ensure security, and support telecommunication services. The course emphasizes practical experience and problem-solving, preparing students to address real-world challenges in server application development..

Format of classes

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

Competencies

GC1. Ability to abstract thinking, analysis and synthesis.
SC5. Ability to develop, improve and use modern software, hardware and software-hardware support for electronic communication and radio devices

Learning outcomes

LO8 – apply general-purpose and specialized programming languages, analytical and simulation modeling packages, as well as software and hardware development tools to solve complex telecommunications and radio engineering problems.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 32, laboratory classes - 32 hours, self-study - 56 hours.

Course prerequisites

"Higher Mathematics", "Theory of Probability", "Programming", "Object-Oriented Programming", "System Software for Telecommunications Systems".

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

Classes are made interactively using multimedia technologies for lecture presentations and online demonstrations of task execution examples. The lecture classes use explanatory-illustrative, reproductive, problem-oriented methods and the method of critical thinking. Training materials are available for students through OneNote (Class Notebook).

Program of the course

Topics of the lectures

Topic 1. Introduction to Server Application Programming:
Understanding the role of server applications in telecommunication systems and their core functionalities.
Topic 2. Server-Side Architectures:
Overview of monolithic, microservices, and serverless architectures and their use cases.
Topic 3. Networking Basics for Server Programming:
Exploring sockets, protocols, and HTTP for communication between clients and servers.
Topic 4. Programming Languages for Server Development:
Examining popular server-side programming languages such as Python, Java, Node.js, and their frameworks.
Topic 5. RESTful API Design:
Principles of designing and implementing RESTful APIs for efficient data exchange.
Topic 6. Authentication and Authorization:
Implementing secure user authentication and role-based access control mechanisms.
Topic 7. Working with Databases:
Integrating relational and NoSQL databases for efficient data storage and retrieval.
Topic 8. Real-Time Communication:
Building real-time applications using WebSockets and other real-time protocols.
Topic 9. Error Handling and Logging:
Best practices for managing server errors, logging, and debugging in server applications.
Topic 10. Security in Server Applications:
Exploring common security threats and implementing strategies to mitigate them, such as encryption and firewalls.
Topic 11. Load Balancing and Scalability:
Techniques for optimizing server performance under high traffic loads.
Topic 12. Testing and Deployment:
Automated testing, CI/CD pipelines, and deploying applications on cloud platforms.

Topic 13. Introduction to Containers and Docker:
Understanding containerization and using Docker for server application deployment.
Topic 14. Monitoring and Performance Optimization:
Utilizing monitoring tools to analyze server performance and optimize resource utilization.
Topic 15. Serverless Computing and Cloud Integration:
Building and deploying serverless applications using platforms like AWS Lambda or Google Cloud Functions.
Topic 16. Case Studies in Server Application Development:
Analyzing real-world examples of server applications in telecommunication systems.

Topics of the workshops

Not included

Topics of the laboratory classes

Topic 1. Setting Up a Server Environment:
Configuring a development environment for server application programming.
Topic 2. Building a Basic REST API:
Creating and testing a simple RESTful API using a server-side framework.
Topic 3. Database Integration:
Connecting a server application to a relational database and performing CRUD operations.
Topic 4. Authentication Implementation:
Implementing user login functionality using token-based authentication.
Topic 5. Real-Time Messaging Application:
Building a real-time chat application using WebSockets.
Topic 6. Error Handling and Logging:
Adding error-handling mechanisms and configuring logging for a server application.
Topic 7. API Testing:
Using tools like Postman or automated scripts to test API endpoints for functionality and security.
Topic 8. Deploying a Server Application:
Deploying a server application on a cloud platform like AWS or Heroku.
Topic 9. Load Balancing Configuration:
Setting up a load balancer to distribute traffic across multiple servers.
Topic 10. Dockerizing a Server Application:
Packaging a server application into a Docker container for easy deployment.
Topic 11. Monitoring Application Performance:
Using monitoring tools like Prometheus or New Relic to analyze server metrics.
Topic 12. Implementing Secure Communication:
Adding SSL/TLS encryption to secure client-server communication.
Topic 13. Implementing Webhooks:
Creating a webhook to handle event-driven communication between services.
Topic 14. Serverless Function Deployment:
Developing and deploying a serverless function on AWS Lambda.
Topic 15. Caching and Optimization:
Implementing caching mechanisms like Redis to improve application performance.
Topic 16. Final Project – Full-Stack Application Development:
Combining server and client-side components to build a complete application..

Self-study

Students are encouraged to supplement their learning through independent exploration of server-side programming concepts, frameworks and tools. This includes practicing coding exercises, experimenting with different server architectures, and studying case studies of real-world server applications. Self-study activities such as reading documentation, following online tutorials, and contributing to open-source projects will deepen understanding and prepare students for complex development tasks.

Course materials and recommended reading

1 Node.js Design Patterns: Design and Implement Production-Grade Node.js Applications by Mario Casciaro and Luciano Mammino (2020)
2. Hands-On Microservices with Python: Build Highly Scalable and Robust Systems with Python by Juan Antonio Medina Iglesias (2021)
3. Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems by Heather Adkins and Betsy Beyer (2020)
4. RESTful Web APIs: Services for a Changing World by Leonard Richardson, Mike Amundsen, and Sam Ruby (2021)
5. Flask Web Development: Developing Web Applications with Python by Miguel Grinberg (2020)
6. Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications by Bilgin Ibryam and Roland Huß (2021)
7. Mastering API Architecture: An API-First Approach to Building Modern Applications by James Gough (2022)
8. Docker in Action by Jeff Nickoloff and Stephen Kuenzli (2021)
9. Web Scalability for Startup Engineers by Artur Ejsmont (2021)
10. Distributed Systems: Principles and Paradigms by Andrew S. Tanenbaum and Maarten Van Steen (2022)

Assessment and grading

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

The final grade is made up of 100% assessment results in the form of test (50%) and current assessment (50%).
Breakdown of the grading:
Laboratory work: 30%
Independent work and computational tasks: 20%

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	Date, signature	Head of the department Pavlo PUSTOVOITOV
	Date, signature	Guarantor of the educational program Vitaliy BRESLAVETS