Syllabus
Course Program

Python Frameworks

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
Special (professional), Elective

Semester
6

Language of instruction
English, Ukrainian

Lecturers and course developers

Svitlana Kovalenko
Svitlana.Kovalenko@khpi.edu.ua
Candidate of Engineering Sciences, Associate Professor, Associate Professor of Software Engineering and Management Intelligent Technologies Department

Google Scholar:
https://scholar.google.com/citations?user=jeD1w74AAAAJ&hl
ORCID: https://orcid.org/0000-0001-6770-6778
Scopus: https://www.scopus.com/authid/detail.uri?authorId=57212035934
Web of Science: https://www.webofscience.com/wos/author/record/F-8252-2017)
More about the lecturer on the department's website

General information

Summary

The objective of the discipline is to provide students with the necessary level of knowledge on the development, testing and deployment of Web-based client-server systems, optimizing their performance, setting up database work using Python frameworks; the ability to integrate tools and frameworks of other programming languages (HTML, CSS, Bootstrap, JavaScript) into the created Web application; studying the functionality of frameworks for designing interface tools that will ensure the execution of data queries

Course objectives and goals

Formation of students' system of theoretical and practical knowledge in the development and testing of fully functional Web applications using powerful tools and ready-made components of Python programming language frameworks

Format of classes

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

Competencies

121-К01. Ability to think abstractly, analyze and synthesize.
121-К02. Ability to apply knowledge in practical situations.
121-К05. Ability to learn and master modern knowledge.
121-К06. Ability to search, process and analyze information from various sources.
121-К13. Ability to identify, classify and formulate software requirements.
121-К15. Ability to develop architectures, modules and components of software systems.
121-К19. Knowledge of data information models, ability to create software for data storage, extraction and processing..
121-К22. The ability to accumulate, process and systematize professional knowledge in the creation and maintenance of software and recognize the importance of lifelong learning.
121-К23. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate software development models and approaches.
121-К25. Ability to reasonably choose and master tools for software development and maintenance.
121-К26. Ability to think algorithmically and logically.
122- GC1. Ability to think abstractly, analyze and synthesis.
122- GC2. Ability to apply knowledge in practical situations.
122- GC6. Ability to learn and master modern knowledge.
122- GC7. Ability to search, process and analyze information from various sources.
122- GC8. Ability to generate new ideas (creativity).
122- GC11. Ability to make informed decisions.
122- PC8. Ability to design and develop software using various programming paradigms: generalised, object-oriented, functional, logical, with appropriate models, methods and algorithms of computation, data structures and control mechanisms.
122- PC9. 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 cloud services.
122- PC10. Ability to apply methodologies, technologies and tools to manage the life cycle processes of information and software systems, information technology products and services in accordance with customer requirements.
122- PC12. Ability to ensure the organization of computing processes in information systems for various purposes, taking into account the architecture, configuration, performance indicators of operating systems and system software.

Learning outcomes

121- PLO01. Analyze, 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.
121- PLO02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.
121- PLO08. Be able to develop a human-machine interface.
121- PLO12. Apply effective software design approaches in practice.
121- PLO13. Know and apply methods of developing algorithms, designing software and data and knowledge structures.
121- PLO15. Motivated to choose programming languages and development technologies to solve the problems of creating and maintaining software.
121- PLO18. Know and be able to apply information technologies for data processing, storage and transmission.
121- PLO23. Be able to document and present the results of software development.
122- PLO1. To apply knowledge of the basic forms and laws of abstract and logical thinking, the basics of the methodology of scientific knowledge, forms and methods of extracting, analyzing, processing and synthesizing information in the subject area of computer science.
122- PLO4. To use methods of computational intelligence, machine learning, neural network and fuzzy data processing, genetic and evolutionary programming to solve problems of recognition, forecasting, classification, identification of control objects, etc.

122- PLO10. To use tools for developing client-server applications, design conceptual, logical and physical models of databases, develop and optimize queries to them, create distributed databases, data warehouses and showcases, knowledge bases, including cloud services, using web programming languages.
122- PLO11. Have the skills to manage the life cycle of software, products and services of information technology in accordance with the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, terms of reference, business plan, agreement, contract).
122- PLO20. Develop the architecture of software systems and their individual components in the construction of intelligent control systems in various industries, as well as manage the life cycle processes of software of intelligent control systems.

Student workload

The total volume of the course is 150 hours (5 ECTS credits): lectures - 32 hours, laboratory classes - 32 hours, self-study - 86 hours.

Course prerequisites

The study of this discipline is directly based on: "Fundamentals of Web Development", "Databases", "Basic Python Programming Course", "Advanced Python Programming 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, problem-based learning.

Forms of assessment:
written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1. opic 1. Basic concepts. Types of Python frameworks. Python web frameworks and their basic properties, WSGI, MVC paradigm
The subject and object of the course "Python frameworks". The concept of framework and library, their similarities and differences, callbacks. Types of Python frameworks: GUI frameworks, Web frameworks, machine learning and testing frameworks. Types of Web frameworks: micro frameworks and full-stack frameworks. The MVC template. The main functions of the Flask microframework core. WSGI. Jinja. Virtual environment. Installing Flask. Creating a simple WSGI application using Flask.
Topic 2. Templates and the Jinja templating engine
The request-response loop of web applications. Processing paths in Flask. View functions. Debugging mode. Templates and their display. Passing parameters to templates. Jinja expressions and operators. Filters. Template inheritance, super() block. Links and static files. Using the Bootstrap framework. Error handling.
Topic 3. Web Forms
Creating and sending a form using HTML and JavaScrip. Module flask_wtf. Processing forms in view functions. Redirection (redirect). Pop-up messages. Validation of forms. Saving data received from the user in the csv file.
Topic 4. Databases in Flask
Types of databases supported by Flask. The Flask-SQLAlchemy module. Definition of the model. Relationships. Operations with the database (creating a table, inserting, deleting, changing rows, querying rows). Using databases in view functions. Database migrations. Deploying the application on cloud services.

Topic 5. Introduction to the Django framework
Загальна інформація про Django. Аналіз відмінностей full stack фреймворків від мікрофреймворків. Встановлення фреймворка, створення проєкту. Команди Django. Створення та налагодження URL, шаблонів та застосунків. Запуск першого застосунку.
Topic 6. Getting started with Django
Django objects and models. Creating applications. Deploying a web application. Git and GitHub. Setting up a pythonanywhere environment. Django URLs. Display functions. Similarities and differences between Django and Jinja templating engines. Possibilities of using Jinja for Django applications. Dynamic data in templates. Creating layouts. Creating, editing, processing, validating and storing forms. Bootstrap integration.
Topic 7. Models and working with them
Introduction to models. Django ORM. Declaring models and model fields. Creating relationships between models. Model parameters. Migrations. Working with data (creating, reading, modifying, deleting). Changes in the database schema. Deleting a model. Routing. Function controllers and class controllers. Pagination. Forms associated with models.
Topic 8: The Django administration site
Activating, configuring, and using the Django admin site.
Topic 9: Users and registration
Cookies. Users and authentication. Logging in and out of a user profile. Managing users, permissions and groups. Preparing an access branching subsystem. Working with lists of users and groups. Authentication and service programs. Getting information about the current user. Authorization. Validation of passwords.
Topic 10. REST web services and deployment
Django REST framework. The principles of REST. Diagnostic and debugging tools. Sending emails. Deploying a website on cloud services.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1. Basics of working with the Jinja templating engine
Topic 2. Working with forms in Flask applications
Topic 3: Using third-party APIs in website development
Topic 4. Creating and deploying an online game using Flask
Topic 5. Creating web-oriented applications based on Django for solving linear programming problems
Topic 6. Working with Django models and deploying an application

Self-study

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

Course materials and recommended reading

Key literature

1. Miguel Grinberg. Flask Web Development: Developing Web Applications with Python, O'Reilly Media; 2nd edition, 2018, 474 p.
2. Malhar Lathkar. Building Web Apps with Python and Flask : Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask Framework, 2021, 307 p.
3. Takatomo Honda Flask Web Development from Scratch: Introduction to Developing Web Applications with Python, 2019, 137 p.
4. William S. Vinscent. Django for Professionals: Production websites with Python & Django (Welcome to Django), 2020, 314 p.
5. Kolawole Mangabo. Full Stack Django and React: Get hands-on experience in full-stack web development with Python, React, and AWS, 2023, 432 p.

Additional literature

1. Al Sweigart. Automate the Boring Stuff with Python, 2nd edition: Practical Programming for Total Beginners, No Starch Press, 2020. 901p.
2. Richard L. Halterman. Fundamentals of Python Programming, 2019, 669p.
3. John Canning, Alan Broder, Robert Lafore. Data Structures & Algorithms in Python, Pearson Education, Inc., 2023, 1050p.
4. Rehan Haider. Web API Development with Python: A Beginner's Guide using Flask and FastAPI (Intermediate Python). 157 p.
5. François Voron. Building Data Science Applications with FastAPI: Develop, manage, and deploy efficient machine learning applications with Python. Packt Publishing, 2021. 426 p.

Assessment and grading

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

100% of the final grade consists of the results of the assessment in the form of a credit (40%) and current assessment (60%):
- 6 laboratory works (7% each);
- 2 tests (9% each).

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