

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ЗАТВЕРДЖЕНО

на засіданні кафедри
інформаційних систем.
Протокол № 8 від 19.01.2024 р.

ПОГОДЖЕНО

Проректор з навчально-методичної роботи

Каріна НЕМАШКАЛО



ПРОГРАМУВАННЯ ІНТЕРНЕТ

робоча програма навчальної дисципліни (РПНД)

Галузь знань	12 "Інформаційні технології"
Спеціальність	121 "Інженерія програмного забезпечення"
Освітній рівень	перший (бакалаврський)
Освітня програма	"Інженерія програмного забезпечення"

Статус дисципліни	обов'язкова
Мова викладання, навчання та оцінювання	англійська

Розробник:
к.т.н., с.н.с.

підписано КЕП

Юрій ПАРФЬОНОВ

Завідувач кафедри
інформаційних систем

Дмитро БОНДАРЕНКО

Гарант програми

Олег ФРОЛОВ

Харків
2024

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS

APPROVED

at the meeting of the department
information systems
Protocol № 8 of 19.01.2024

AGREED

Vice-rector for educational and methodical work
Karina NEMASHKALO



INTERNET PROGRAMMING
Program of the course

Field of knowledge **12 "Information technologies"**
Specialty **121 "Software engineering"**
Study cycle **first (bachelor)**
Study program **"Software Engineering"**

Course status **mandatory**
Language **English**

Developer:
PhD (Information technology),
Senior Research Assistant digital signature Yurii PARFONOV

Head of Information systems
Department _____ Dmitrii BONDARENKO

Head of Study Programme _____ Oleg FROLOV

**Kharkiv
2024**

[Handwritten notes in blue ink at the bottom right corner of the page]

INTRODUCTION

Nowadays, the most widespread software systems are those developed using Internet technologies. This requires software engineering specialists to have a clear understanding of the general concepts of Internet programming and to use modern tools for developing the server side of web applications.

"Internet Programming" is a mandatory course that is studied in accordance with the curriculum for the bachelor's degree in the specialty 121 "Software Engineering".

An essential element of successful mastering of the discipline's educational material is the students' independent work with technical literature and modern development environments.

The aim of the course "Internet Programming" is to develop competencies in the use of modern Internet programming tools and technologies for developing the server side of web applications.

The objectives of the course are to master modern Internet programming technologies; improve skills in developing server-side web applications and using development environments.

The object of the course is the development of Internet applications.

The subject of the course is modern technologies for developing server-side web applications.

The learning outcomes and competencies formed by the course are defined in Table 1.

Table 1

Learning outcomes and competencies formed by the course

Learning outcomes	Competencies
LO5	SC14
LO7	GC2, GC5, SC10, SC13
LO8	GC5, GC6
LO14	SC4
LO15	GC2, GC10, GC13
LO16	GC7
LO17	GC2, GC7, SC10, SC12, SC13

where, LO5. Know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modeling for software development.

LO7. To know and apply in practice the fundamental concepts, paradigms, and basic principles of functioning of language, technological and computing tools of software engineering.

LO8. Be able to develop a human-machine interface.

LO14. Apply in practice software tools for domain analysis, design, testing, visualization, measurement, and documentation of software.

LO15. Motivated to choose programming languages and development technologies to solve the problems of creating and maintaining software.

LO16. Have skills in team development, approval, design, and release of all types of program documentation.

LO17. Be able to apply the methods of component-based software development.

GC2. Ability to apply knowledge in practical situations.

GC5. Ability to learn and master modern knowledge.

GC6. Ability to search, process and analyze information from various sources.

GC7. Ability to work in a team.

SC4. Ability to formulate and ensure software quality requirements in accordance with customer requirements, terms of reference and standards.

SC10. Ability to accumulate, process and systematize professional knowledge of software development and maintenance and recognize the importance of lifelong learning.

SC12. Ability to carry out the system integration process, apply change management standards and procedures to maintain the integrity, overall functionality, and reliability of the software.

SC13. Ability to reasonably choose and master tools for software development and maintenance.

SC14. Ability to think algorithmically and logically.

COURSE CONTENT

Content module 1: Using Internet programming tools in Python

Topic 1: Introduction to the course

1.1. Distributed software systems. Software components. Architecture of distributed software systems. Middleware.

1.2. Support for Internet protocols and related technologies in Python.

Topic 2. Using Internet protocols in Python

2.1. Using SMTP, POP3, IMAP, etc. protocols. Support for Internet protocols in Python.

Topic 3: Using Web Scraping in Python

3.1 Fundamentals of Web Scraping. Libraries for Web Scraping.

Content module 2: Basics of developing the server side of web applications based on Django framework

Topic 4. The basics of Django framework

4.1 Web applications. The MVC template. Web frameworks. Server-side web frameworks.

4.2. General information about Django framework. Implementation of the MVT template.

4.3. Installing Django. The structure of Django web application. Routing HTTP requests.

Topic 5. Django models

5.1. General information about database access. Object-relational data mapping. The concept of Django Object Relational Mapping.

5.2. Introduction to models. Class of the model. Types of model fields. Relationship between models. Setting up models. Metaclasses. Database migrations.

Topic 6: Types and templates in Django

6.1. Development of types. Mapping URLs and views.

6.2. Template structure. Data exchange between a template and a view.

6.3. Basics of the template language. Using tags in templates. Partial templates. Inheritance of templates.

Topic 7. Forms

7.1. Form classes. The process of form processing in Django. Form fields. Validation. Standard forms for models.

Topic 8: Class-based views

8.1. General information. Use of generic views. Inheritance of views. Using mixins.

Topic 9: Using the admin site

9.1. General information about the administrator site. Registration of models. Creating a super user. Setting up the interface of the administrator site.

Topic 10. Sessions

10.1. General information about sessions. Using sessions. Saving session data. Setting up sessions.

Topic 11: Authorization and authentication

11.1. Authorization and authentication system in Django. Activating authentication. Creating users and groups.

Topic 12: Internationalization and localization

12.1. Internationalization in the application source code, template code, and URL. The basics of localization in Django.

The list of laboratory studies in the course is given in Table 2.

Table 2

The list of laboratory studies

Name of the topic	Content
Topic 2. Laboratory session 1	Using Internet protocols in Python
Topic 3. Laboratory session 2	Using Web Scraping in Python
Topic 4. Laboratory session 3	Django basics
Topics 5 - 7. Laboratory session 4	Using databases in Django-based web applications
Topic 8: Laboratory session 5	Using class-based views
Topic 9: Laboratory session 6	Using the Django admin site
Topics 10, 11. Laboratory session 7	Using sessions, authentication, and authorization in Django
Topic 12: Laboratory session 8	Using internationalization and localization in Django

The list of self-studies in the course is given in Table 3.

Table 3

List of self-studies

Name of the topic	Content
Topics 1 - 12	Study of lecture material
Topics 1 - 12	Preparing for laboratory classes
Topics 1 - 12	Preparing for the final exam

The number of hours of lectures, laboratory classes, and hours of self-study is given in the technological card of the course.

TEACHING METHODS

In the process of teaching the course, to acquire certain learning outcomes, and activate the educational process, it is envisaged to use such teaching methods as:

Verbal (lecture (Topics 1, 3, 4 - 11), problem-based lectures (Topics 2, 12)).

Visual (demonstration (Topics 1 - 12)).

Practical (laboratory studies, interactive distance learning, coaching, presentation, role-playing games (Topics 1 - 12); case studies (Topics 11, 12)).

FORMS AND METHODS OF ASSESSMENT

The University uses a 100-point cumulative system for assessing the learning outcomes of students.

Current control is carried out during lectures and laboratory classes and is aimed at checking the level of readiness of the student to perform a specific job and is evaluated by the amount of points scored:

- for courses with a form of semester control as an exam: maximum amount is 60 points; the minimum required is 35 points.

The final control includes current control, and exam.

Semester control is carried out in the form of a semester exam.

The final grade in the course is determined:

– for disciplines with a form of exam, the final grade is the amount of all points received during the current control and the exam grade.

During the teaching of the discipline, the following control measures are used:

Current control: defense of laboratory work (50 points), control work (10 points).

Semester control: Exam (40 points)

More detailed information on the assessment system is provided in technological card of the course.

An example of an exam card and assessment criteria.

Example of an exam card

Task 1

Develop a Django-based Web application to work with the Student data model, whose objects are stored in SQLite database. The class of the data model explicitly defines attributes, such as Surname, Faculty, Course.

The Web application must contain:

1. An HTML page with a Django form that allows you to enter the student's name, faculty, and course in its separate fields and send this data to the Function-based View (FBV).
2. FBV should write the entered data to the database, using the developed service class; read it from the database by calling the appropriate methods of the service class; in the HTTP response of the server - transfer the extracted data to another HTML page for displaying (each value should be on a separate line).
3. A service class that contains methods for adding data to the database and reading data from the database using the Django ORM API.

Task 2

Create a superuser to access admin site of the web application from Task 1. Register the Student data model of the application to use it in conjunction with the admin site. Perform programmatic customization of the admin site web interface to display a custom web page title instead of "Django administration" and to filter the records displayed in the list by last name and faculty.

Assessment criteria

The exam card contains two complex heuristic tasks. The results of each task are assigned a rating from 0 to 6 according to the following scale:

6	The task has been completed in full. The program works correctly. The source code of the program meets the requirements specified in the assignment.
5	The task is completed. The program works correctly, but it was developed in violation of one of the requirements specified in the assignment.
4	The task is mostly completed. The program works, but it is designed in violation of two of the requirements specified in the assignment.
3	The assignment was completed with significant drawbacks. The program works, but one of the requirements specified in the assignment is not fulfilled, or three of the requirements are implemented with violations
2	The task is practically not completed. The program runs, but two of the requirements specified in the task are not met, or four of the requirements are implemented with violations
1	The task is not completed. The program runs and partially meets the task formulation, but more than two of the requirements specified in the task are not met, or more than four of these requirements are implemented in violation. Or the program does not start or crashes, but there is a program code developed by the student that meets the task formulation.
0	The task is not completed. There is no program or the program does not contain source code developed by the student, or the program does not meet the task formulation, or the program has obvious technical signs of non-independent development.

Failure to comply with or material violation of any of the general requirements for the program reduces the performance rating for each such task by one (1).

General requirements for programs

1. The user interface should correspond to the task formulation and contain text labels (messages) in English related to the input and output of basic and auxiliary data, their validation, etc. The use of transliteration is prohibited.
2. Compliance with PEP 8 - Style Guide for Python Code (<https://www.python.org/dev/peps/pep-0008/>).
3. The source code of each Python class must be in a separate file.

Task 1 of the exam card is mandatory. It is scored up to 90% of the maximum grade. Task 2 of the exam card is optional and scored up to 10% of the maximum grade. The maximum grade on a 40-point scale for each task is determined as follows:

Rating of the results of the task	Grade for Task 1	Grade for Task 2
6	36	4
5	33 - 35	3
4	30 - 32	2
3	26 - 29	1
2	20 - 25	0
1	1 - 19	
0	0	

RECOMMENDED LITERATURE

Main

1. Mele A. Django by Example / A. Mele. - Packt Publishing Limited, Birmingham 2020 - 568 p.
2. Vincent W. Django for Beginners: Build websites with Python and Django / W. Vincent. - WelcomeToCode, 2023 - 349 p.

Additional

3. Feldroy D. Two Scoops of Django 3.X: Best Practices for Django / D. Feldroy. - Two Scoops Press, 2020 - 477 p.

Information resources

4. The Python Standard Library [Electronic resource]. - Access mode: <https://docs.python.org/3/library/index.html>.
5. Interactive Python tutorial [Electronic resource]. - Access mode: <https://www.learnpython.org/>.
6. Django documentation [Electronic resource]: <https://docs.djangoproject.com/en/5.0/>.
7. Django forum [Electronic resource]. - Access mode: <https://forum.djangoproject.com/>.

8. DjangoSites: Shiny websites, powered by Django [Electronic resource]. - Access mode : <https://djangosites.org/>.

9. Django Tutorial: The Local Library website [Electronic resource]: https://developer.mozilla.org/en/docs/Learn/Server-side/Django/Tutorial_local_library_website.

10. Django Admin Cookbook [Electronic resource]. - Access mode: <https://books.agiliq.com/projects/django-admin-cookbook/en/latest/>.

11. Django Tutorial [Electronic resource]. - Access mode: <https://www.w3schools.com/django/>.

12. Парфьонов Ю. Е., Колгатін О. Г. Вибір служби вебхостингу для застосунків на базі фреймворку Django / Ю. Е. Парфьонов, О. Г. Колгатін // Вісник Харківського національного автомобільно-дорожнього університету: збірник наукових праць. - Харків: ХНАДУ, 2022. – Вип. 96. – С. 66 – 70. – Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/27384>