

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 12:07:09  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Информационно-управляющие  
системы»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

---

**РАБОЧАЯ ПРОГРАММА**

дисциплины

**«ОСНОВЫ БЭКЕНД-РАЗРАБОТКИ НА PYTHON Ч.2»**

для подготовки бакалавров

по направлению

09.03.02 «Информационные системы и технологии»

по профилю

**«Информационно-управляющие системы»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н., доцент Заславский М.М.

ассистент Иванов Д.В.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## **1 СТРУКТУРА ДИСЦИПЛИНЫ**

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ
Общая трудоемкость (ЗЕТ)	3
Курс	3
Семестр	6

### **Виды занятий**

Электронные лекции (акад. часов)	34
Электронные практические (академ. часов) (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	1
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

### **Вид промежуточной аттестации**

Дифф. зачет (курс)	3
--------------------	---

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ОСНОВЫ БЭКЕНД-РАЗРАБОТКИ НА PYTHON Ч.2»**

Дисциплина является второй частью курса, в рамках которого обучающиеся получают знания, умения и навыки программирования на языке Python, изучают основы работы с IDE, получают навыки работы с базами данных и важнейшими библиотеками. Это позволит обучающимся писать скрипты для парсинга и автоматизации задач, работать с данными и создавать приложения на Python.

#### **SUBJECT SUMMARY**

#### **«BASICS OF PYTHON BACKEND DEVELOPMENT PART 2»**

The discipline is the second part of the course, in which students gain knowledge, skills, and programming skills in Python, learn the basics of working with the IDE, and gain skills in working with databases and critical libraries. This will allow students to write scripts for parsing and automating tasks, work with data, and create applications in Python.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является получение теоретических знаний в области разработки приложений для мобильных платформ, а также формирование практических умений и навыков по применению полученных знаний для решения задач профессиональной деятельности.
2. Задачами дисциплины являются приобретение знаний, умений и навыков в области бэкенд-разработки на языке Python.
3. Обучающиеся получают знания о работе в облачных сервисах, взаимодействии фронтенда и бэкенда, работе на сервере и настройке окружения.
4. В результате освоения дисциплины приобретаются умения обработки сообщений и команд, использования инструментария для тестирования API, запуска приложений в docker-контейнерах.
5. Обучающиеся приобретают навыки развертывания проектов на удаленных серверах, автоматического тестирования проектов.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Алгоритмы и структуры данных»
2. «Основы бэкенд-разработки на Python ч.1»

и обеспечивает подготовку выпускной квалификационной работы.

### **3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
ПК-0.1	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
ПК-0.2	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
ПК-0.3	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

## **4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **4.1 Содержание разделов дисциплины**

#### **4.1.1 Наименование тем и часы на все виды нагрузки**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>ЭЛек, ач</b>	<b>ЭПр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение	1			
2	Введение в API	6	6		7
3	Django REST Framework	6	6		8
4	Настройка API-сервиса	6	6		8
5	Работа на сервере	6	8		8
6	Docker для Python-разработчика	8	8		8
7	Заключение	1		1	
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе				108/3

#### **4.1.2 Содержание**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение	Содержание второй части курса
2	Введение в API	Что такое API. Работа с внешними API. Форматы обмена данными. Взаимодействие программ по сети. API First. Архитектура REST. REST: ресурсы, эндпоинты и HTTP-методы. Исследование запросов. Механизмы авторизации, протокол OAuth 2.0. Обработка входящих сообщений и команд. Отправка изображений. Кнопки вместо текста
3	Django REST Framework	Django Rest Framework. Финальный проект спринта: CRUD для Yatube. REST API: Проектирование. Инструментарий для тестирования API. Преобразование форматов. Сериализаторы. View-функции API. View-классы API. Высокосеты и роутеры. Сериализаторы для связанных моделей. Сериализаторы: дополнительные настройки. Регулярные выражения. Высокосеты. Расширенные возможности. Аутентификация по токену. JWT + Djoser

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
4	Настройка API-сервиса	Права, лимиты запросов и фильтрация ответов в DRF. Взаимодействие фронтена и бэкенда. Проверка прав: Permissions. Throttling: ограничение количества запросов. Пагинация в API. Фильтрация, сортировка и поиск. Добавление фотографий. Упрощаем взаимодействие с фронтендом. Multi Page Application vs Single Page Application. Документация для API.
5	Работа на сервере	Работа на сервере, настройка окружения. Введение в DevOps. Где развертывать приложение. Подключение к удалённому серверу. Внеплановая задача и первое развертывание. Server Gateway Interface: выбор, установка и управление. Веб-и обратный прокси-сервер Nginx: установка и настройка. Доменное имя. Шифрование. SSL-сертификат. Мониторинг доступности и сбор ошибок.
6	Docker для Python-разработчика	Docker: приложение в коробке. Автоматизация. Развертывание на сервере. Виртуальные машины и контейнеры. Установка Docker. Запуск приложений в docker-контейнерах. Упаковка проекта в Docker-образ. Сервер Gunicorn и Docker volume. DockerHub. Архитектура приложения в Docker. PostgreSQL и Docker Network. Docker-compose: сборка проекта. Docker-compose: Nginx и статика. Публикация проекта в Docker на сервере. [Docker Hub] Автоматизация деплоя: CI/CD. GitHub Actions. Первый workflow. Workflow для CI: линтер и тесты. Workflow для CD: сборка образов и перезапуск контейнеров.
7	Заключение	Общие выводы по изложенному материалу. Связь с последующими дисциплинами траектории.

## 4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

## 4.3 Перечень практических занятий

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Django REST Framework	10
2. Настройка API-сервиса	12
3. Docker	12
Итого	34

#### **4.4 Курсовое проектирование**

Курсовая работа (проект) не предусмотрены.

#### **4.5 Реферат**

Реферат не предусмотрен.

#### **4.6 Индивидуальное домашнее задание**

Индивидуальное домашнее задание не предусмотрено.

#### **4.7 Доклад**

Доклад не предусмотрен.

#### **4.8 Кейс**

Кейс не предусмотрен.

#### **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единий логин и пароль). Каждую неделю будет доступна новая тема курса: видеолекции, кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал

последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение нескольких недель после его окончания.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	15
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
Самостоятельное изучение разделов дисциплины	8
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	8
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференциированному зачету, экзамену	8
<b>ИТОГО СРС</b>	<b>39</b>

## **5 Учебно-методическое обеспечение дисциплины**

### **5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины**

<b>№ п/п</b>	<b>Название, библиографическое описание</b>	<b>К-во экз. в библ.</b>
<b>Основная литература</b>		
1	Баланов А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов, 2024. -312 с. -Текст : электронный.	неогр.
2	Чернышев С. А. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON 2-е изд., пер. и доп. : Учебник для СПО / А. Чернышев С., 2025. -349 с. -Текст : непосредственный.	неогр.
<b>Дополнительная литература</b>		
1	Информатика. Введение в Python : электрон. учеб. пособие / К. В. Кринкин [и др.], 2020. -1 эл. опт. диск (CD-ROM). -Текст : электронный.	неогр.
2	Янцев В. В. Web-программирование на Python : учебное пособие для вузов / В. В. Янцев, 2024. -180 с. -Текст : электронный.	неогр.

### **5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины**

<b>№ п/п</b>	<b>Электронный адрес</b>
1	Руководство по работе Docker в Python <a href="https://uproger.com/gajd-po-rabote-docker-v-python-ot-ustanovki-do-slozhnyh-primerov/">https://uproger.com/gajd-po-rabote-docker-v-python-ot-ustanovki-do-slozhnyh-primerov/</a>

### **5.3 Адрес сайта курса**

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=23639>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

Для дисциплины «Основы бэкенд-разработки на Python ч.2» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### **Дифференцированный зачет**

<b>Оценка</b>	<b>Количество баллов</b>	<b>Описание</b>
Неудовлетворительно	0 – 7	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	8 – 11	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	12 – 13	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Отлично	14 – 15	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

## **Особенности допуска**

К дифференциированному зачету допускаются студенты, показавшие прогресс в ЭОС Moodle (не менее 80%) и выполнившие 3 практические работы на оценку не менее "удовлетворительно".

Дифференцированный зачет проводится в виде тестирования в электронной образовательной среде Moodle. Тест содержит 15 вопросов с развернутыми ответами. Каждый вопрос теста оценивается в 1 балл в случае правильного ответа. Оценка дифференциированного зачета выставляется в зависимости от количества баллов, полученных за итоговый тест следующим образом:

"Неудовлетворительно" -менее 8 баллов

"Удовлетворительно" -от 8 до 11 баллов

"Хорошо" -от 12 до 13 баллов

"Отлично" -от 14 до 15 баллов.

## **6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине**

### **Вопросы к дифф.зачету**

<b>№ п/п</b>	<b>Описание</b>
1	Работа с внешними API
2	Архитектура REST. REST: ресурсы, эндпоинты и HTTP-методы
3	Механизмы авторизации, протокол OAuth 2.0.
4	Обработка входящих сообщений и команд.
5	Django Rest Framework
6	Инструментарий для тестирования API
7	View-классы API. Вьюсеты и роутеры
8	Аутентификация по токену
9	Throttling: ограничение количества запросов
10	Multi Page Application vs Single Page Application
11	Документация для API
12	Введение в DevOps
13	Server Gateway Interface: выбор, установка и управление.
14	Веб-и обратный прокси-сервер Nginx: установка и настройка
15	SSL-сертификат.
16	Docker: приложение в коробке.

17	Виртуальные машины и контейнеры.
18	Архитектура приложения в Docker.
19	PostgreSQL и Docker Network
20	Workflow для CD: сборка образов и перезапуск контейнеров

## Вариант теста

1. Какой заголовок HTTP-ответа указывает тип содержимого?
  1. Content-Type
  2. Accept
  3. Content-Length
  4. Encoding

Ответ: Content-Type

2. Какой модуль Python используется для кодирования URL?
  1. urllib.parse
  2. requests.utils
  3. http.encode
  4. url.encode

Ответ: urllib.parse

3. Какой командой создаётся новый проект Django?
  1. django-admin startapp
  2. django-admin startproject
  3. python manage.py newproject
  4. python django createproject

Ответ: django-admin startproject

4. Какой файл в Django отвечает за маршрутизацию URL?
  1. models.py
  2. views.py
  3. urls.py
  4. settings.py

Ответ: urls.py

5. Какой ORM-метод Django используется для получения всех объектов модели?

1. .get()
2. .filter()
3. .all()
4. .select()

Ответ: .all()

6. Какой декоратор Django используется для ограничения доступа к представлению только аутентифицированным пользователям?

1. @login\_required
2. @auth\_required
3. @permission\_required
4. @secure\_view

Ответ: @login\_required

7. Какой командой применяются миграции в Django?

1. python manage.py makemigrations
2. python manage.py migrate
3. python manage.py update\_db
4. python manage.py db\_upgrade

Ответ: python manage.py migrate

8. Какой инструмент используется для тестирования API?

1. Postman
2. Selenium
3. PyTest
4. JUnit

Ответ: Postman

## **Образцы задач (заданий) для контрольных (проверочных) работ**

### **Пример практической работы**

Разверните и активируйте виртуальное окружение, реализуйте:

1. Кастомные страницы для ошибок
  - Подключите к проекту и настройте кастомные страницы для ошибок 403 CSRF, 404 и 500
2. Работу с пользователями
  - Подключите к проекту пути для работы с пользователями
  - Переопределите шаблоны для каждой подключённой страницы.
  - Создайте страницу с формой для регистрации пользователей.
  - Создайте страницу пользователя. На ней должны отображаться:
    - информация о пользователе (доступна всем посетителям)
    - публикации пользователя (доступны всем посетителям),
    - ссылка на страницу редактирования профиля для изменения имени, фамилии, логина и адреса электронной почты (доступна только хозяину аккаунта),
    - ссылка на страницу изменения пароля (доступна только хозяину аккаунта).

Переопределять встроенную модель пользователя не требуется.

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### **6.3 График текущего контроля успеваемости**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
1	Введение в API Django REST Framework	
2		
3		
4		
5		Практическая работа
6	Настройка API-сервиса Работа на сервере	
7		
8		
9		
10		
11	Docker для Python-разработчика	Практическая работа
12		
13		
14		
15		
16		
17		Практическая работа

### **6.4 Методика текущего контроля**

#### **на лекционных занятиях**

Текущий контроль включает в себя:

- контроль прогресса в ЭОС Moodle (не менее 80%).

#### **на практических занятиях**

- контроль прогресса в ЭОС Moodle (не менее 80%)

В процессе обучения по дисциплине студент обязан выполнить 3 практические работы в ЭОС Moodle.

Студент должен показать: понимание постановки задачи, подхода к ее решению, ход решения, выбор тех или иных методик решения задачи. Преподаватель определяет глубину понимания теоретического материала, который лежит в основе решения задачи практической работы, а также самостоятельность ее выполнения.

**Критерии оценивания практических работ:**

«отлично» - вопрос раскрыт полностью, задача решена правильно

«хорошо» - вопрос раскрыт не полностью, задача решена частично

«удовлетворительно» - в ответе на вопрос имеются существенные ошибки; задача не решена или решена неправильно, ход решения правильный

«неудовлетворительно» - отсутствует ответ на вопрос или содержание ответа не совпадает с поставленным вопросом, задача не решена, ход решения неправильный

### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на лекционных, практических занятиях по методикам, описанным выше.

## **7 Описание информационных технологий и материально-технической базы**

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше

## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>