**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**ОУ “КЫРГЫЗСКО-КОРЕЙСКИЙ КОЛЛЕДЖ”**

«Утверждаю»

Директор ККК

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бейшеналиева Г.Б.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

**УЧЕБНО – МЕТОДИЧЕСКИЙ КОМПЛЕКС**

По дисциплине: «Server Programming»

Для студентов направления: Computer Science

дневной формы обучения

Учебно-методический комплекс разработан преподавателем \_\_\_\_\_\_Нармухамедовым Р.Т.

**Бишкек 2024г.**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**ОУ “КЫРГЫЗСКО-КОРЕЙСКИЙ КОЛЛЕДЖ”**

«Утверждаю»

Директор ККК

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бейшеналиева Г.Б

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

**РАБОЧАЯ ПРОГРАММА**

По дисциплине: «Server Programming»

Для студентов направления: Computer Science

Дневной формы обучения

Всего часов по учебному плану 90

из них:

Лекций………………...............................................16

Практических занятий..............................................32

СРС………………….................................................42

Итоговый контроль – экзамен

Рабочая программа разработана преподавателем \_\_\_\_\_\_\_\_\_\_\_ Нармухамедовым Р.Т.

**Бишкек 2024г.**

**1.Введение (пояснительная записка)**

Дисциплина "Server Programming" фокусируется на разработке REST API с использованием Django. Студенты изучают проектирование RESTful-сервисов, работу с Django REST Framework, моделирование данных, сериализацию, маршрутизацию запросов и тестирование API. Курс охватывает аутентификацию, авторизацию и оптимизацию производительности, позволяя студентам уверенно разрабатывать и развертывать API для различных приложений.

Основными критериями отбора тем и методов для изучения являются:

 Актуальность: Темы должны соответствовать современным требованиям и трендам в области программирования.

 Практическая значимость: Изучаемые методы должны быть применимы в реальных проектах и задачах, что способствует развитию практических навыков.

 Постепенность сложности: Темы должны быть организованы так, чтобы последовательно увеличивать уровень сложности и углублять знания студентов.

 Междисциплинарность: Темы должны интегрироваться с другими дисциплинами, что позволяет студентам видеть связь между различными областями знаний.

**2. Цели освоения дисциплины**

Рабочая программа учебной дисциплины " Server Programming " ориентирована на достижение следующих целей:

1. **Обеспечение** студентов глубокими знаниями основ разработки серверных приложений с использованием Django.
2. **Развитие** умений работы с Django REST Framework для создания эффективных веб-сервисов
3. **Обучение** методам тестирования, документирования и оптимизации API.
4. Подготовка студентов к решению реальных задач в области разработки серверных приложений.

Применение знаний о серверном программировании позволяет создавать эффективные и масштабируемые веб-приложения, обеспечивать взаимодействие между клиентами и серверами, а также реализовывать сложные бизнес-логики. Эти навыки важны для разработки API, которые служат основой для мобильных и веб-услуг. Кроме того, знание серверного программирования помогает оптимизировать производительность приложений, гарантировать безопасность данных и упрощать интеграцию с другими системами. Специалисты могут работать над проектами, требующими обработки больших объемов данных, и разрабатывать решения, которые обеспечивают высокую доступность и надежность сервисов.

В результате освоения дисциплины студент должен знать:

 Понимание основ REST API и принципов работы серверного программирования

 Способность создавать REST API с использованием Django и Django REST Framework.

 Умение настраивать маршрутизацию и обрабатывать запросы для различных HTTP-методов.

 Знание методов аутентификации и авторизации для обеспечения безопасности API.

В результате освоения дисциплины студент должен уметь:

 Проектировать и реализовывать RESTful API с использованием Django и Django REST Framework

 Работать с моделями данных, сериализовать и десериализовать информацию для обмена между клиентом и сервером.

 Реализовывать методы аутентификации и авторизации для обеспечения безопасности API.

 Тестировать и документировать разработанные API для улучшения их функциональности и удобства использования.

**3. Методы изучения дисциплины.**

Изучение данной дисциплины предполагает проведение:

Лекций – 16 часов,

Практических занятий - 32 часа,

СРС – 42 часа.

Промежуточная аттестация производится согласно учебному плану, рабочему учебному плану и графику учебного процесса.

Проведение итогового контроля по завершении курса-экзамен.

**Содержание дисциплины *Fundamentals of Algorithms***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование темы** | **Лекционные занятия** | **Практические занятия** | **СРС** |
| **1 модуль** | | | | |
| **1.** | Введение в серверное программирование и REST API | 2 | 2 | 4 |
| **2.** | Django как серверный фреймворк | 1 | 2 | 2 |
| **3.** | Django REST Framework: Введение и настройка | 1 | 2 | 4 |
| **4.** | Работа с сериализаторами | 1 | 2 | 2 |
| **5.** | Работа с моделями и базой данных | 1 | 2 | 4 |
| **6.** | Аутентификация и авторизация в REST API | 1 | 2 | 2 |
| **7.** | Обработка ошибок и валидация данных | 1 | 2 | 4 |
| **8.** | Рубежный контроль: Тест | 2 | | |
| **2 модуль** | | | | |
| **9.** | Документирование и тестирование REST API | 2 | 2 | 2 |
| **10.** | Структура данных. Список, методы списков, кортеж, словарь, набор. | 1 | 2 | 4 |
| **11.** | Деплой и развертывание API | 1 | 2 | 2 |
| **12.** | Оптимизация и кэширование API | 1 | 2 | 4 |
| **13.** | Мониторинг и логирование REST API | 1 | 2 | 2 |
| **14.** | Версионирование и управление изменениями API | 1 | 2 | 4 |
| **15.** | Проектирование и разработка полноценного REST API, объединение всех изученных методов и принципов. | 1 | 2 | 2 |
| **16.** | Рубежный контроль: Тест | 2 | | |

**График проведения рейтингового контроля**

|  |  |
| --- | --- |
| **Наименование тем** | **Сроки сдачи** |
| Практическая работа №1  Практическая работа №2  Практическая работа №3  Практическая работа №4  Практическая работа №5  Практическая работа №6 | До 8 недели |
| **Модуль 1** | 8-я неделя |
| Практическая работа №7  Практическая работа №8  Практическая работа №9  Практическая работа №10  Практическая работа №11  Практическая работа №12 | До 16 недели |
| **Модуль 2** | 16-я неделя |
| **Экзамен** | 17-я неделя |

**Оценка знаний, умений и навыков**

В рабочем учебном плане предусмотрено проведение СРС, организационного контроля, рубежного контроля и итогового экзамена. Организационный контроль – оценка посещаемости студентов за период учебного года. Рассматривается по окончании 1 и 2 модулей за семестр, одновременно для всех студентов.

Рубежный контроль – оценка знаний по освоению пройденного материала. Проводится по окончании 1 и 2 модулей, одновременно для всех студентов.

Итоговый экзамен проводится по окончанию семестра в период экзаменационной недели.

Сумма общего балла – СРС, организационного (ОК), промежуточного (РК) и итогового контроля (ИК) исчисляется следующим путем:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 модуль | | | 2 модуль | | | Текущий контроль | ИК | Общий балл |
| ОК 1 | СРС 1 | РК 1 | ОК 2 | СРС 2 | РК 2 |
| 10 | 10 | 10 | 10 | 10 | 10 | 60 | 40 | 100 |

Баллы распределяются следующим образом:

**«Отлично»**

- А+ (95-100)

- А (90-94)

- А- (85-89);

**«Хорошо»**

- В+(80-84)

- В (75-79)

- В (70-74);

**«Удовлетворительно»**

- С+ (67-69)

- С (64-66)

- С- (60-63);

**«Неудовлетворительно»:**

- D (40-59)

- F (<40)

Сумма баллов всех дисциплинарных модулей и итогового экзамена должна составлять минимальных - 60, а максимальных – 100.

**Учебно-методические материалы**

1. \*\*"Django for Beginners"\*\*  William S. Vincent

2. \*\*"Django RESTful Web Services"\*\*  Gaston C. Hillar

3. \*\*"Django 3 By Example"\*\*  Antonio Mel

4. \*\*"Two Scoops of Django 3"\*\*  Daniel and Audrey Roy Greenfeld

5. \*\*"Django for Professionals"\*\*  William S. Vincent

6. \*\*"Django Unleashed"\*\*  Andrew Pinkham

7. \*\*"Building RESTful Python Web Services with Flask"\*\*  Gustavo Schmitt

8. \*\*"Python Crash Course"\*\*  Eric Matthes (   -)

9. \*\*"Flask Web Development"\*\*  Miguel Grinberg (   Django)

10. \*\*"RESTful Web APIs"\*\*  Leonard Richardson  Sam Ruby