

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 24.06.2026 06:57:39
Уникальный идентификатор документа:
e3a68f3eaa1e62674b54f4998099d3d6bfd4076

**Оценочный материал для промежуточной аттестации по дисциплине
«Основы WEB-инжиниринга» 3 семестр**

Квалификация выпускника	бакалавр
Направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Безопасность информационных систем и технологий <i>наименование</i>
Форма обучения	очная
Кафедра разработчик	Информатики и вычислительной техники <i>наименование</i>
Выпускающая кафедра	Информатики и вычислительной техники <i>наименование</i>

Типовые задания для контрольной работы

Задание 1: Проектирование веб-приложения

Задание:

Опишите архитектуру простого веб-приложения для управления библиотекой. Включите в описание компоненты, такие как:

Бэкенд (язык программирования, фреймворк, база данных).

Фронтенд (язык программирования, фреймворк или библиотека).

API (REST или GraphQL).

Основные функции (добавление книг, управление пользователями, выдача и возврат книг).

Требования:

Опишите каждый компонент архитектуры.

Приведите схему взаимодействия между клиентом и сервером.

Задание 2: HTML и CSS

Задание:

Создайте веб-страницу, которая содержит:

Заголовок с названием проекта (H1).

Навигационное меню.

Блок контента с текстом и изображением.

Футер с контактной информацией.

Используйте HTML для разметки и CSS для стилизации страницы.

Требования:

Для стилей используйте внешний CSS-файл.

Добавьте адаптивный дизайн, чтобы страница корректно отображалась на мобильных устройствах.

Задание 3: Создание простого RESTful API

Задание:

Разработайте простое RESTful API для управления задачами (TODO-список) с использованием

языка программирования, с которым вы знакомы (например, Python, JavaScript, PHP).

Приложение должно поддерживать следующие операции:

Получение всех задач.

Добавление новой задачи.

Изменение существующей задачи.

Удаление задачи.

Требования:

Опишите каждую из конечных точек (endpoint) API.

Укажите, как будет обрабатываться формат данных (например, JSON).

Задание 4: SQL – Создание базы данных

Задание:

Создайте SQL сценарий для создания базы данных для вашей системы управления библиотекой, которая включает следующие таблицы:

Книги (id, название, автор, год издания, жанр).

Пользователи (id, имя, контактная информация).

Выдачи (id, id_книги, id_пользователя, дата_выдачи, дата_возврата).

Требования:

Укажите типы данных для каждого поля.

Определите первичные и внешние ключи, если это необходимо.

Задание 5: Основы тестирования веб-приложений

Задание:

Опишите основные подходы к тестированию веб-приложений, включая:

Юнит-тестирование.

Интеграционное тестирование.

Тестирование пользовательского интерфейса (UI).

Требования:

Объясните важность каждого подхода в разработке.

Приведите примеры инструментов, которые можно использовать для каждого типа тестирования.

Типовые вопросы к экзамену:

1. Философские аспекты проблемы систем искусственного интеллекта (возможность существования, безопасность, полезность).
2. Технология разработки web-приложений (понятие Клиент, Сервер, Сервер приложений, Common Gateway Interface (CGI))
3. Скриптовые технологии разработки web-приложений на стороне сервера
4. Скриптовые технологии разработки web-приложений на стороне клиента
5. Структура протокола HTTP
6. HTML, основные понятия, схема, принципы организации информации.
7. Взаимодействие web-приложения с СУБД
8. AJAX - общие принципы работы
9. CSS - основные понятие, принцип каскадности
10. Принципы аутентификации web-приложений (cookies, session)
11. Принципы работы сети Internet (архитектура, топология, общие принципы маршрутизации)
12. Взаимодействие пользователя и web-сервера: GET и POST запросы. Принципы передачи параметров.
13. Стэк протоколов TCP/IP, адресация ресурса в сети. (Физический адрес, IP адрес, порт, протокол, URL)
14. Язык программирования Python: общие принципы построения программы (блок, подключение модулей)
15. Python: организация циклов, ветвлений.
16. Python: операции со строками (конкатенация, выделение подстроки)
17. Python: операции с числами (деление, остаток от деления, инкремент)
18. Flask - framework, принцип организации приложения.
19. Jinja2 - способы построения шаблонов.
20. CSS-framework - на примере Bootstrap (что это такое, и для чего может использоваться)