1. Что такое фреймворк?

Структура, предоставляющая набор инструментов для разработки приложений.

1. Что такое фреймворк операционной системы?

Набор библиотек и инструментов, предоставляемых системой для упрощения разработки приложений под конкретную систему.

1. Что такое программная платформа?

Это совокупность программных средств( операционная система, фреймворки, библиотеки, инструменты разработки), которые обеспечивают среду для создания, запуска и управления приложениями.

1. Что такое .NET?

Кроссплатформенная программная платформа для разработки десктопных, мобильных и веб-приложений.

1. Что такое ASP.NET CORE?

Кроссплатформенный фреймворк с открытым исходным кодом для разработки web-приложений на .NET.

1. Что означает, что ASP.NET CORE является кроссплатформенной?

Позволяет создавать и запускать приложения на разных операционных системах.

1. Что такое NuGet?

Менеджер пакетов для .NET, позволяет разработчикам создавать и использовать библиотеки, предоставлять к ним общий доступ

1. Для чего нужна утилита dotnet?

Утилита dotnet — это командная строка для управления проектами .NET. С её помощью можно:

* Создавать новые проекты и решения.
* Собирать и запускать приложения.
* Управлять пакетами и зависимостями через интеграцию с NuGet.
* Публиковать приложения для развертывания.

Она предоставляет единый интерфейс для различных задач разработки и развертывания приложений на платформе .NET.

1. Что такое CoreCLR?

CoreCLR (Core Common Language Runtime) — это основная реализация среды выполнения языка общего назначения для .NET Core. Она отвечает за выполнение управляемого кода, предоставляя такие функции, как сборка мусора, управление памятью, компиляция в машинный код (JIT), а также обеспечение безопасности и изоляции кода.

1. Что такое CoreFX?

CoreFX — это **библиотека, интегрированная в .** **NET**. Среди её компонентов: System. Collections, System.IO, System.

1. Как называется встроенный в ASP.NET CORE http-сервер?

Встроенный в ASP.NET Core HTTP-сервер называется Kestrel. Он является кроссплатформенным, высокопроизводительным веб-сервером, используемым по умолчанию для обработки HTTP-запросов в приложениях ASP.NET Core.

1. Стадии (этапы) подготовки и запуска **ASPA.**

Основные этапы подготовки и запуска ASP.NET Core приложения включают:

1. **Инициализация**: Создание объекта WebApplicationBuilder, который настраивает конфигурацию приложения, сервисы и другие параметры.
2. **Конфигурация сервисов**: Добавление необходимых сервисов в контейнер зависимостей через метод ConfigureServices.
3. **Построение приложения**: Создание объекта WebApplication с помощью метода builder.Build(), который объединяет все настройки и подготавливает приложение к запуску.
4. **Настройка конвейера обработки запросов (Middleware)**: Определение последовательности компонентов middleware, которые будут обрабатывать входящие HTTP-запросы.
5. **Запуск приложения**: Вызов метода app.Run(), который запускает приложение и начинает прослушивать входящие запросы.

13. Выясните последнюю устойчивую версию ASP.NET CORE.