1. Дайте определение понятию «Интернет».

**Интернет** - это глобальная сеть компьютеров, соединенных между собой по всему миру, с использованием стандартизированных протоколов и технологий, таких как TCP/IP, для обмена информацией и ресурсами.

1. Дайте определение понятию «Служба Интернет».

**Служба Интернет** обычно относится к различным сервисам и ресурсам, доступным через Интернет. Это может включать веб-сайты, электронную почту, стриминговые платформы, социальные сети и другие онлайн-сервисы.

1. Дайте определение понятию «Узел сети Интернет».

**Узел сети Интернет** - это устройство или компьютер, подключенный к Интернету и способный передавать или получать данные через сеть.

1. Дайте определение понятию «клиент-серверное приложение».

**Клиент-серверное приложение** - это архитектурный подход в сетевом программировании, в котором клиентский компьютер (клиент) обращается к серверу для получения данных, ресурсов или услуг, которые сервер предоставляет.

1. Дайте определение понятию «сетевой протокол».

**Сетевой протокол** - это набор правил и стандартов, определяющих способ обмена данными между устройствами в компьютерной сети. Примерами сетевых протоколов являются HTTP, TCP, IP, FTP

1. Перечислите основные свойства протокола HTTP.

Основные свойства протокола **HTTP** (Hypertext Transfer Protocol) включают:

* Базируется на клиент-серверной архитектуре.
* Основанный на тексте, читаемый человеком.
* Без сохранения состояния (stateless).
* Использует методы запроса (GET, POST, PUT, DELETE) для взаимодействия с ресурсами.
* Работает поверх протокола TCP/IP.

1. Перечислите состав информации, пересылаемой в HTTP-запросе.

**HTTP-запрос** обычно включает в себя:

* Метод (GET, POST, PUT, DELETE) - тип запроса.
* URL - адрес ресурса.
* Заголовки (headers) - метаданные запроса.
* Тело (body) - данные, отправляемые на сервер (не всегда присутствует).

1. Перечислите состав информации, пересылаемой в HTTP-ответе.

**HTTP-ответ** обычно содержит:

* Статусный код (status code) - указывает на результат выполнения запроса (например, 200 для успешного запроса).
* Заголовки (headers) - метаданные ответа.
* Тело (body) - данные, возвращаемые сервером (например, HTML страница или JSON).

1. Дайте определение понятию «web-приложение».

**Web-приложение** - это программное приложение, которое выполняется на сервере и предоставляет интерфейс и функциональность через веб-браузеры клиентов. Пользователи могут взаимодействовать с такими приложениями через интернет.

1. Дайте определение понятиям «frontend» и «backend».

**Frontend** и **backend** - это две части web-приложения:

* **Frontend** отвечает за пользовательский интерфейс и взаимодействие с пользователем в браузере. Это включает в себя HTML, CSS и JavaScript для отображения и работы с данными.
* **Backend** отвечает за обработку запросов от клиентов, взаимодействие с базой данных, бизнес-логику и предоставление данных клиенту. Это может быть реализовано с использованием различных языков программирования и фреймворков.

1. Дайте определение понятию «кроссплатформенное приложение».

**Кроссплатформенное приложение** - это приложение, спроектированное и разработанное таким образом, чтобы оно могло работать на разных операционных системах и устройствах без существенных изменений или с минимальными модификациями.

1. Изобразите и поясните общую схему web-приложения.

Общая схема web-приложения включает в себя клиентов (браузеры) и сервер, которые обмениваются данными через Интернет. Клиент отправляет HTTP-запросы на сервер, а сервер возвращает HTTP-ответы с данными. Frontend обрабатывает отображение данных и пользовательский интерфейс, а backend управляет логикой и обработкой запросов.

1. Назовите основные технологии разработки серверных кроссплатформенных приложений.

Основные технологии разработки серверных кроссплатформенных приложений включают Node.js, Python (с использованием фреймворков Django или Flask), Ruby on Rails, PHP и Java (с использованием фреймворков Spring или Java EE).

1. Поясните понятие «асинхронная операция».

**Асинхронная операция** - это операция, которая выполняется без блокировки выполнения других операций. В асинхронной модели выполнения, операции выполняются параллельно, и программа может продолжать работу без ожидания завершения операции.

1. Поясните принцип выполнения асинхронного запроса с помощью объекта XMLHTTPRequest и Fetch.

Асинхронный запрос с помощью объекта **XMLHTTPRequest** и **Fetch** позволяет браузеру отправлять запросы на сервер и продолжать выполнение других задач, не блокируя интерфейс. Когда ответ от сервера будет получен, будет вызвана соответствующая функция обратного вызова для обработки ответа.

1. Поясните основное назначение сервера NODE.JS.

**Node.js** - это среда выполнения JavaScript на сервере, которая позволяет разработчикам создавать высокопроизводительные и масштабируемые сетевые приложения. Основное назначение Node.js - обработка асинхронных операций и создание серверов.

1. Перечислите основные свойства сервера NODE.JS.

Основные свойства сервера **Node.js**:

* Основан на событийной модели и асинхронном вводе/выводе.
* Поддерживает однопоточную, неблокирующую обработку запросов.
* Использует JavaScript для серверной разработки.
* Позволяет создавать высоконагруженные приложения с использованием платформы V8 JavaScript Engine.