

## **Ответы на вопросы к защите лабораторной работы №7.**

### **Взаимодействие клиента и сервера**

1. Клиент-серверная архитектура — это модель, где клиент (например браузер) запрашивает данные, а сервер их обрабатывает и возвращает ответ. Основные части: клиентское приложение, серверная часть, база данных и сетевые соединения между ними.
  2. В вебе чаще всего используют HTTP/HTTPS для обычных запросов, WebSocket для постоянного соединения, FTP для загрузки файлов и TCP/IP как основу сетевого взаимодействия. HTTPS защищает данные шифрованием.
  3. RESTful API — способ организации веб-сервисов, где URL представляют ресурсы, а HTTP-методы (GET, POST и др.) определяют действия. Например, GET /api/users получает список пользователей, а POST /api/users создает нового.
  4. Сессии хранят данные пользователя на сервере, а куки — маленькие метки в браузере, помогающие серверу узнавать клиента между запросами. Например, куки хранят ID сессии для авторизации.
  5. Аутентификация проверяет кто пользователь (по паролю или токену), авторизация — что ему разрешено. Для защиты используют шифрование (HTTPS), проверку данных и специальные токены вместо паролей в запросах.
  6. При ошибках сервер отправляет код состояния: 200 — успех, 400 — ошибка в запросе, 500 — проблема на сервере. В теле ответа обычно пишут подробности в формате JSON.
  7. CORS разрешает браузеру делать запросы к другим доменам, если сервер явно указал разрешенные источники в заголовках ответа. Без этого браузер блокирует межсайтовые запросы.
  8. Оптимизация запросов ускоряет работу сайта. Например, сжатие данных уменьшает их размер, кэширование избегает повторных загрузок, а CDN быстрее доставляет статику.
  9. Кэширование сохраняет данные для повторного использования. Браузер кэширует статику по указанию сервера (заголовки Cache-Control), а сервер может кэшировать частые запросы в памяти или Redis.
  10. Статическое содержимое (картинки, CSS) не меняется и отдается как есть. Динамическое (персональные рекомендации, результаты поиска) генерируется сервером под каждый запрос с учетом текущих данных.
- (Никаких списков, только текст с примерами. Если нужно что-то перефразировать — скажи.)