Vue.js

1. Что такое Vue.js и для каких целей он используется?
2. Каковы основные преимущества Vue.js перед другими фреймворками?
3. Что такое Vue instance и как он используется?
4. Как работает система реактивности в Vue.js?
5. Что такое директивы в Vue.js и какие существуют основные директивы?
6. Как использовать условный рендеринг в Vue.js?
7. Что такое компоненты в Vue.js и как они создаются?
8. Как осуществляется передача данных между компонентами в Vue.js?
9. Что такое слоты в Vue.js и как они используются?
10. Как работает система маршрутизации в Vue.js?
11. Что такое Vuex и для чего он используется?
12. Как управлять состоянием приложения с помощью Vuex?
13. Какие методы жизненного цикла компонента существуют в Vue.js?
14. Как осуществляется интеграция Vue.js с серверными технологиями?
15. Какие инструменты используются для тестирования приложений на Vue.js?
16. Как в Vue.js реализуется динамический рендеринг компонентов?
17. Как использовать миксины в Vue.js?
18. Что такое директива v-model и как она используется?
19. Как в Vue.js обрабатываются пользовательские события?
20. Какие стратегии оптимизации производительности существуют в Vue.js?
21. Как использовать фильтры в Vue.js для форматирования данных?
22. Что такое Vue Router и как он настраивается?
23. Как в Vue.js реализовать глобальное управление состоянием без Vuex?
24. Какие возможности предоставляет Vue CLI?
25. Как в Vue.js реализовать серверный рендеринг (SSR)?

AngularJS

1. Что такое AngularJS и в чем его отличие от Angular?
2. Какие основные особенности AngularJS?
3. Что такое директивы в AngularJS и как они используются?
4. Как в AngularJS реализуется двустороннее связывание данных (two-way data binding)?
5. Что такое scope в AngularJS и как он используется?
6. Как в AngularJS создаются и используются контроллеры?
7. Что такое сервисы в AngularJS и как они применяются?
8. Как работает маршрутизация в AngularJS?
9. Как осуществляется внедрение зависимостей (Dependency Injection) в AngularJS?
10. Какие существуют способы оптимизации производительности приложений на AngularJS?
11. Какие инструменты используются для тестирования в AngularJS?
12. Какие проблемы могут возникнуть при миграции с AngularJS на более новые версии Angular?
13. Как в AngularJS обрабатываются формы и валидация данных?
14. Какие паттерны проектирования часто используются в AngularJS?
15. Как AngularJS обеспечивает безопасность приложений?
16. Как в AngularJS реализуется анимация элементов?
17. Как использовать $watch, $digest и $apply для управления данными?
18. Что такое директива ng-repeat и как она используется?
19. Как в AngularJS реализовать кастомные директивы?
20. Какие лучшие практики существуют для написания чистого и поддерживаемого кода в AngularJS?
21. Как в AngularJS обрабатываются AJAX-запросы?
22. Как использовать $routeProvider для настройки маршрутов?
23. Какие есть способы оптимизации загрузки приложения на AngularJS?
24. Как реализовать ленивую загрузку (lazy loading) в AngularJS?
25. Какие особенности работы с формами в AngularJS следует учитывать?

ReactJS

1. Что такое ReactJS и для каких задач он наиболее подходит?
2. Каковы ключевые особенности и преимущества ReactJS?
3. Что такое JSX и как он используется в ReactJS?
4. Как в ReactJS создаются компоненты?
5. Что такое props и state в ReactJS и как они используются?
6. Как работает механизм подъема состояния (lifting state up) в ReactJS?
7. Что такое контекст (Context) в ReactJS и как его использовать?
8. Как в ReactJS реализуется управление формами?
9. Что такое хуки (Hooks) в ReactJS и какие существуют основные хуки?
10. Как использовать эффекты (useEffect) в ReactJS?
11. Как осуществляется маршрутизация в ReactJS приложениях?
12. Какие стратегии оптимизации производительности существуют в ReactJS?
13. Как интегрировать ReactJS с другими библиотеками или фреймворками?
14. Какие инструменты используются для тестирования ReactJS приложений?
15. Как управлять глобальным состоянием в приложениях ReactJS?
16. Как в ReactJS использовать компоненты высшего порядка (HOC)?
17. Как реализовать мемоизацию компонентов в ReactJS?
18. Что такое React Fiber и как он улучшает производительность?
19. Как использовать Portals в ReactJS?
20. Как в ReactJS реализовать Error Boundaries?
21. Какие паттерны проектирования полезны при работе с ReactJS?
22. Как реализовать международную поддержку (i18n) в ReactJS?
23. Как в ReactJS работать с асинхронными запросами?
24. Какие нововведения в ReactJS были представлены в последних версиях?
25. Как использовать React с TypeScript для повышения производительности и безопасности кода?

WebSockets

1. Что такое WebSockets и для каких задач они наиболее подходят?
2. Как устанавливается соединение WebSocket?
3. Какие преимущества WebSockets перед традиционным HTTP?
4. Как обеспечивается безопасность в WebSockets?
5. Какие ограничения и недостатки имеют WebSockets?
6. Как интегрировать WebSockets с фреймворками на стороне сервера?
7. Как обрабатывать сообщения WebSocket на клиенте?
8. Какие сценарии использования WebSockets наиболее распространены?
9. Как управлять состоянием соединения WebSocket?
10. Какие стратегии масштабирования существуют для WebSocket-серверов?
11. Какие библиотеки и инструменты поддерживают работу с WebSockets?
12. Какие проблемы могут возникнуть при использовании WebSockets и как их решать?
13. Как тестировать приложения, использующие WebSockets?
14. Как WebSockets взаимодействуют с протоколами HTTP/2 и HTTP/3?
15. Какие альтернативы WebSockets существуют для реализации реального времени в веб-приложениях?
16. Как реализовать авторизацию и аутентификацию в приложениях с WebSockets?
17. Какие паттерны проектирования полезны при работе с WebSockets?
18. Как обеспечить совместимость WebSockets с различными браузерами?
19. Какие существуют библиотеки для работы с WebSockets в Node.js?
20. Как реализовать шифрование данных в WebSockets?
21. Какие механизмы восстановления соединения существуют в WebSockets?
22. Как использовать WebSockets в мобильных приложениях?
23. Какие есть лучшие практики для уменьшения задержек в WebSockets?
24. Как WebSockets влияют на нагрузку сервера?
25. Какие инструменты и методы используются для мониторинга и отладки WebSockets?