Q1

**Расскажите, чем, на ваш взгляд, отличается хорошее клиентское приложение от**

**плохого с точки зрения**

** пользователя:**

Хорошее приложение обладает  
-Адаптивностью: можно открыть на любом устройстве, в браузере или в приложении, оффлайн или онлайн.  
-Функционалом: выполняет те функции и запросы, которые нужны пользователю – быстро, без артефактов, падений и сбоев.  
- Приятным UI: смена дневной и ночной темы, большие кнопки, понятный дизайн, ненавязчивая начинка без раздражающих уведомлений и мигающих реклам.   
-Доступностью: смена шрифтов, легкая версия, возможность писать/ звонить в поддержку, быстрая загрузка страниц (ssr, lazy-loading)

 **менеджера проекта:**

Хорошее приложение можно  
- Расширять: добавлять/убирать фичи, наращивать клиентскую базу и не терять в производительности и быстродействии  
- Тестировать: в приложении пишутся необходимые тесты, позволяющие ускорить разработку и снизить технический долг.  
- Рефакторить: код, который смогут понять и улучшать/обновлять новые люди, приходящие в проект. Выполнение Код-ревью, постановка сроков по задачам.  
- Поддерживать: снизить количество падений, версионирование кода, защита, смена кадров без потери производительности приложения.

** дизайнера:**

При работе с Хорошим приложением есть  
- Постоянное сотрудничество с верстальщиком и фронтенд-разработчиком  
- Набор базовых компонентов (цвета, шрифты, темы) и форм (например storybook)  
- Подробное ТЗ  
- Руководство грамотного менеджера, осуществляющего связь с заказчиком, рассмотрение и утверждение макетов

** верстальщика:**  
В Хорошем приложении есть  
- Постоянное сотрудничество с дизайнером, фронтенд-разработчиком, seo - специалистом  
- Внутренний регламент верстки и готовая начальная сборка на Gulp/ Webpack  
- Автоматизация верстки, готовый стартовый шаблон

** серверного программиста:**  
при работе с Хорошим приложением есть  
- Архитектурные решения от Head of Team/TeamLead  
- Постоянное сотрудничество с коллегами (бэкенд, фронтенд-разработчики, девопс и т.д.)  
- Внутренний регламент разработки бэка, логики, сервера, базы данных, деплоя  
- Возможность использовать удобные платные инструменты, технологии: железо, поддержка, трекеры, облако и т.д.

Q2

Опишите основные особенности разработки крупных многостраничных сайтов,

функциональность которых может меняться в процессе реализации и поддержки.

Расскажите о своем опыте работы над подобными сайтами: какие подходы,

инструменты и технологии вы применяли на практике, с какими проблемами

сталкивались и как их решали.

Q3

При разработке интерфейсов с использованием компонентной архитектуры часто

используются термины **Presentational Сomponents** и **Сontainer Сomponents**. Что

означают данные термины? Зачем нужно такое разделение, какие у него есть плюсы и

минусы?

Q4

Как устроено наследование в JS? Расскажите о своем опыте реализации JS-

наследования без использования фреймворков.

Q5

Какие библиотеки можно использовать для написания тестов end-to-end во

фронтенде? Расскажите о своем опыте тестирования веб-приложений.

Q6

Вам нужно реализовать форму для отправки данных на сервер, состоящую из

нескольких шагов. В вашем распоряжении дизайн формы и статичная верстка, в

которой не показано, как форма должна работать в динамике. Подробного описания,

как должны вести себя различные поля в зависимости от действий пользователя, в

требованиях к проекту нет. Ваши действия?

Q7

Расскажите, какие инструменты помогают вам экономить время в процессе

написания, проверки и отладки кода.

Q8

Какие ресурсы вы используете для развития в профессиональной сфере? Приведите

несколько конкретных примеров (сайты, блоги и так далее).

Какие ещё области знаний, кроме тех, что непосредственно относятся к работе, вам

интересны?

Q9

Расскажите нам немного о себе и предоставьте несколько ссылок на последние

работы, выполненные вами.