## Level I

**>Q1**

**>**

**\*Хорошее приложение - это...\***

Для пользователя - удобство пользования (как бы каламбурно ни звучало) и быстрота работы приложения (скорость нынче в моде).

Для менеджера - возможность внедрять (и, конечно, продать) новые фичи.

Для дизайнера - когда дизайн чётко решает поставленные задачи, и ещё внешний вид при этом опрятный.

Для верстальщика - когда у приложения все элементы едины в рамках одного проекта, когда можно переиспользовать те или иные части вёрстки, а то и вовсе руководствоваться стайлгайдом.

Для программиста - приложение с хорошо продуманной логикой, когда известны все сущности и зависимости между ними.

---

**>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.**

Для подобных проектов важно придерживаться единых стандартов внутри команды. Зачастую, крупные многостраничные сайты проектируются, утверждаются, и разрабатываются довольно долго. В редких случаях может смениться человек на той или иной позиции. Важно создать такую инфраструктуру, при которой изменение функциональности в процессе разработки не приведёт к ситуации "ааааа, всё перевёрстывать с нуляяяяя". Разработчикам надо использовать проверенные методики и инструменты, постараться не тестировать новый функционал на таком крупном проекте, без согласования руководства и других участников команды. Верстальщикам надо придерживаться именования классов, одного препроцессора, одной системы сборки, и т. д. Можно создать стайлгайд и пинать каждый раз дизайнеров, чтобы они тоже старались ему следовать, и, по-возможности, переиспользовали элементы.

Но это теория. На практике, бывают сложности. Бывали проблемы нехватки времени, когда нет возможности задокументировать код, отрефакторить, и, как следствие, другому разработчику было сложно вникнуть в происходящее. Проблема всегда индивидуально внутри каждой команды, должна решаться сообща.

Вообще, в моей бывшей команде я создал заготовку для вёрстки (аналог boilerplate, но конкретно под наши нужды) - это решало многие проблемы, которые бывали ранее.

---

**>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?**

Сontainer Сomponents ---компонент в котором принимают и работают с данными

Presentational Сomponents ---компонент в котором отображены данные

Этот подход помогает разделить получение и отображение данных.  
О плючах и минусах сказать не могу, потому что не достаточно компетентен в этом вопросе (пока что не достаточно)

**>Q4**

**>Как устроено наследование в JS? Расскажите о своем опыте реализации JS-наследования без использования фреймворков.**

Классическое наследование - через прототип. Сталкивался только в процессе обучения.

---

**>Q5**

**>Какие библиотеки можно использовать для написания тестов end-to-end во фронтенде? Расскажите о своем опыте тестирования веб-приложений.**

Мне это предстоит изучить.

---

**>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?**

Зависит от договорённостей внутри команды, на мой взгляд. Если эти договорённости позволяют некую свободу действий, и разработчик не лишён "чувства прекрасного" (в состоянии сам сделать приятные элементы интерфейса и анимации), то и незачем напрягать дизайнера. Я бы реализовал простой но опрятный внешне вариант (принцип 20\80), и показал его коллегам-дизайнерам, вместе обсудили бы, и, либо приняли, либо доработали.

---

**>Q7**

**>Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.**

Пока не использовал ничего такого, наверное потому что серьёзной работы не было.

**>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?**

Интерент, книги нет(не люблю читать книги ☺).

Области знаний: нравится 3d моделирование, уверенно чувствую себя в программе blender

---

**>Q9**

**>Расскажите нам немного о себе и предоставьте несколько ссылок на последние работы, выполненные вами.**

Я обучаемый, начинающий разработчик, никаких конкретных задач и работ не делал, только то что в процессе самообучения. Поэтому показать нечего. ☺

--