Q1: Расскажите, чем, на ваш взгляд, отличается хорошее клиентское приложение от плохого с точки зрения:

1. Пользователя

Хорошее приложение должно быстро работать, предоставляя при этом пользователю требуемый функционал, а также иметь дружелюбный по отношению к пользователю интерфейс, не перегруженный яркими или плохо сочетающимися цветами.

1. Верстальщика

Хорошее приложение можно легко визуально разбить на различные блоки и элементы, чтобы без труда создать сетку, каркас страницы для её последующего заполнения HTML-элементами.

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?

Компоненты-контейнеры как правило содержат в себе сложную логику (изменения стейта, обработка данных, отслеживание своего жизненного цикла и т.д.), в то время как презентационные компоненты служат для отрисовки того, что «положат» в их пропсы компоненты-контейнеры.

Такое разделение нужно для более явного разделения внутренней логики приложения и его отображения – контейнеры вычисляют, что будет видеть пользователь, а презентационные компоненты определяют, как это будет выглядеть. При таком подходе проще организовать структуру приложения, предварительно определив, какой компонент за что будет отвечать, также становится проще отлаживать код. Однако не всегда можно с лёгкостью определить, сколько логики «достанется» определенным компонентам, и в таких случаях на проработку архитектуры приложения может понадобиться дополнительное время.

Q4: Как устроено наследование в JS? Расскажите о своем опыте реализации JS-наследования без использования фреймворков.

Наследование в JS происходит при помощи прототипов. Прототип – это объект, который есть у каждого объекта в JS, на него ссылается свойство \_\_proto\_\_. При создании нового объекта его свойства будут взяты из прототипа. Таким образом, чтобы унаследовать один объект от другого, достаточно в прототип первому вписать ссылку на нужный объект.

Я применял прототипное наследование в нескольких учебных проектах. В одном из них два схожих по структуре объекта наследовались из более абстрактного родительского «класса», часть методов которого позже перегружалась в «детях», а часть наследовалась напрямую, без изменений.

Q5: Какие библиотеки можно использовать для написания тестов end-to-end во фронтенде? Расскажите о своем опыте тестирования веб-приложений.

У меня очень скромный опыт тестирования, большая часть которого была получена в результате разработки тестового задания для Вашей фирмы. В данном случае я использовал интегрированный в Create React App jest с добавлением enzyme для тестирования React-компонентов.

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?

Скорее всего, придётся определять обязательность тех или иных полей «на глаз», защитив их от неправильного ввода данных и попыток отправить форму в таком виде на сервер.

Q7: Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.

DevServer в Webpack’e позволяет значительно экономить время, динамично обрабатывая изменения в коде/стилях и позволяя их увидеть без ручных перезагрузок страницы. В Chrome очень удобные инструменты разработчика, позволяющие легко менять ширину страницы (для сравнения у Firefox инструменты расположены внизу страницы и позволяют менять только её высоту, что весьма бесполезно) для тестирования отображения на разных расширениях, удобная консоль. Также можно упомянть различные встроенные возможности редактора Atom (богатый набор плагинов, интеграция с Git и т.д.).

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?

На таких сайтах, как Хабрахабр или Medium есть немало интересных статей по веб-разработке, stackoverflow помогает «чинить» код и иногда подкидывает идеи для реализации того или иного функционала. Также я нашёл несколько каналов на youtube, позволяющие знакомиться с новыми технологиями в относительно короткий срок (The Net Ninja, Traversy Media).

Мне интересен космос, музыка 70-х и 80-х, паззлы, фантастические произведения, видеоигры.

Q9: Расскажите нам немного о себе и предоставьте несколько ссылок на последние работы, выполненные вами.

За последний год я старался как можно активнее изучить веб-разработку и всё, что с ней связано, создав несколько проектов, которые можно увидеть здесь:

https://github.com/OskarZeta

Очень хочу применить свои знания на практике, устроившись к Вам на работу. Я спокойный и немногословный человек, пытающийся найти своё место в жизни. Надеюсь на плодотворное сотрудничество.