**Ответы на вопросы FunBox**

Q1.

* Пользователь. Для пользователя важно удобство использования и никакие дефекты не должны его отвлекать и мешать ему, все таки пользователь заходит получить информацию или выполнить какое другое действие, а не смотреть на дефекты верстки, если в верстке будет некорректно что-то отображаться или работать, если что-то будет перекрывать другой элемент, к которому должен быть доступ, то, соответственно, и отзыв об опыте использования будет не лучшим
* Менеджер проекта. Я думаю, что менеджеру проекта важны адаптивность, кроссбраузерность, валидность верстки, использование современных технологий, дабы проект работал быстро, использование специальных тегов (h1…, nav и другие). Ему также важно, чтобы на следующих этапах разработки было меньше проблем. Он также выступает в роли пользователя, то есть все должно работать и отображаться без дефектов
* Дизайнер. В первую очередь, я думаю, что верстальщик должен сохранить идею и концепцию дизайнера, если есть какие-то другие идеи и предложения, то можно предложить это дизайнеру и менеджеру, если дадут добро, то вносить свои идеи, а менять по своему хотению
* Верстальщик. Для верстальщика самое плохое, что может быть – это нижний скролл :) Должна использоваться методология БЭМ, это позволяет избегать неочевидных и непонятных названий классов, конфликтов, да и работать и поддерживать такой код намного проще.
* Клиентский программист. Для клиентского программиста показателем хорошей верстки, я думаю, будет являться продуманность и хорошая структура верстки, чтобы можно было менять только какие-то отдельные части компонента. Удобно было бы перемещаться по компоненту, отслеживать какую-то определенную логику. К примеру: есть модальное окно с формой, и после отправки формы и получение ответа с сервера, надо поменять форму на окно “спасибо”. Чтобы не перерисовать полностью все модальное окно или не выдумывать какие то непонятные штуки, внутри окна должен быть специальный контейнер для контента, где мы просто будет менять контент.
* Серверный программист. Буду исходить из своего опыта натяжки верстки на CMS. Для серверного программиста важно удобство интеграции верстки с бэкендом. К примеру: если это фоновые картинки, то они должны быть подключены через атрибут style, а не через css. Да и должно быть понятно, где ему искать тот или иной компонент, иначе если будет каша в коде, то как минимум верстальщик будет икать)

Q2. При верстке многостраничных сайтов важно выносить повторяющиеся элементы на разных страницах в отдельные файлы (шапки, футеры, сайдбары и пр.). Благодаря этому будет проще вносить изменения в верстку, т.к. теперь надо будет менять, например шапку, не на всех страницах, а только в одном файле. Если есть некоторые отличия, к примеру опять шапки, на разных страницах, то лично я делал так: определял текущую страницу и присваивал шапке определенный класс

Q3. При адаптивной верстке, для корректного отображения ресурса на всех устройствах, необходимо подгонять верстку под текущий размер экрана: изменять размеры шрифтов, отступы, размеры картинок, потому что, если даже верстка не поехала, пользователю должно быть удобно получать информацию. Для написания адаптива можно использовать, к примеру в google, специальную панель разработчика, где можно задавать размер экрана и проверять как выглядит ресурс. После желательно проверить на необходимом устройстве (телефон, планшет). Я не нашел нормального бесплатного ресурса для проверки сайта на IOS, поэтому приходилось просить у знакомых телефон, что очень неудобно. Верстка под IOS бывает затруднительна, т.к. сафари поддерживает не все css стили и js функции. Например, сафари не поддерживает функцию scrollIntoView, которую надо было реализовать самостоятельно, когда я использовал Gulp. Но сейчас, думаю, babel справиться с этой задачей сам. Или был случай, когда в сафари, по непонятным мне причинам, добавлял нижний скролл. Для устранения этой проблемы я добавлял обертку для всей верстки со стилем overflow: hidden, что решало эту проблему. Еще сафари “приближает” окно при нажатии на форму. Эта проблема решается указанием в мета теге с именем viewport - maximum-scale=1

Q4. Кончено же королем в ускорении написания html, является emmet) Это кончено же еще и препроцессоры. В pug удобно реализовывать компонентый подход, приходится меньше писать кода, но в нем, на мой взгляд, легко запутаться, поэтому тут больше привлекает html с emmet. Но в pug однозначно есть свои плюсы. Scss, Sass, Less позволяют быстрее писать стили. Очень удобно использовать вложенности, переменные и миксины. А так для отладки и проверки кода из плагинов ничего не использую, кроме Bracket Pair Colorizer. Можно писать юнит-тесты (использовал Jest) и использовать TypeScript для проверки типов значений

Q5. Я с такой проблемой не сталкивался. Но я думаю, что для начала надо увидеть или попросить, чтобы описали, что отображается не так. После этого уже может быть понятно, в чем заключается проблема. Конечно, было бы прекрасно попробывать поправить ошибку в панели разработчика, но в сафари, наверно, это можно будет сделать, используя Mac, или если есть специальные ресурсы, то через них, но мне не удалось найти. Может плохо искал) В любом случае, можно использовать интернет, скорее всего, с такой проблемой уже кто-то сталкивался, и можно найти решение этой проблемы

Q6. Я бы попросил дизайнера либо объяснить этот момент, либо отрисовать это. Но, конечно, лучше, если бы дизайнер отрисовал поведение элемента

Q7. Для изучения чего-то нового мне больше нравится смотреть видео, но иногда информации больше и лучше в интернете. Сейчас смотрю канал Владилена Минина, у него много полезного контента для Frontend разработчика. Нравится канал от 0 до 1. Много подчерпнул оттуда знаний по верстке. Канал Гоши Дударя тоже неплохой. Раньше много видео там смотрел. JavaScript изучал на learn.javascript.ru

Я готов развиваться в JavaScript и в верстке, но больше меня привлекает js-программирование. Мне больше нравится разрабатывать какую-то логику

На данный момент, я хотел бы изучить глубже JS, различные фреймворки, работу JS на бэкенде. Пока такие планы. Может, когда достигну хорошего уровня в этом направлении, захочу приобрести еще какой-то навык

Q8. Меня зовут Владислав и мне 18 лет. Примерно 1.5 года занимаюсь версткой и натяжкой на CMS. Поработал на фрилансе, выполнил несколько проектов. В общей сложности у меня есть 10 работ в портфолио по верстке, 4 из которых я выполнил на фрилансе. До этого пробывал себя в разных направлениях: писал игры на C++ (был первый мой язык), разрабатывал игры на js, писал приложение на desktop на electron.js, пытался разрабатывать под мобилку на Java. Пробывал себя везде по чуть-чуть. Полгода назад меня заинтересовал язык JavaScript. Изучил Webpack, TypeScript, написал свой плагин (range slider) на TypeScript с архитектурой MVC, изучил React, Redux, познакомился с REST API (но я все равно немного не понимаю, что это означает, есть пробелы)

Последний сайт, который я делал на фрилансе, и, на мой взгляд, самый качественный -> <https://github.com/Vlgdev/AnDesign>  
Вот ссылка на реальный проект –> <https://andesign.pro/>

Здесь вы можете посмотреть просто функционал, который я реализовывал. У меня просто не все исходники сохранились, и могут быть недочеты в верстке. Так же делал на фрилансе, это был первый проект на фрилансе –> <https://github.com/Vlgdev/visual-up>  
Ссылка на реальный проект -> <https://visual-up.ru/>

Это плагин, который я разрабатывал -> <https://github.com/Vlgdev/plugin>

Так же можете посмотреть некоторые работы по React в моем репозитории, но они были сделаны по обучающим курсам