**Q1:**

• С точки зрения пользователя, плохая верстка может выражаться в неудобном, беспорядочном расположении элементов страницы. Например, могут возникнуть сложности с навигацией (не понятно, где меню и как оно работает), индикацией (не понятно, выбран элемент или нет) и восприятием информации (броские цвета, разное направление текста, отсутствие мобильной версии).

• С точки зрения менеджера проекта: несоответствие изначальному техническому заданию, плану веб-страницы и функций, которые она должна была выполнять.

• С точки зрения дизайнера: неверно подобранная цветовая схема страницы, некорректное отображение значков, кнопок и др.элементов страницы.

• С точки зрения верстальщика: мусорные теги, лишние классы, безобразная структура проекта и грязный код (с излишней или недостающей табуляцией).

• С точки зрения клиентского программиста: неверные теги элементов (вместо заголовка просто текст с отдельным стилем, лишние ссылки внутри head и др.).

• С точки зрения серверного программиста: использование устаревших технологий/неверно подобранный стек технологий.

\_\_\_\_\_

**Q2:**

При верстке крупных и многостраничных проектов важно делать акцент на их расширяемость. Например, более детальное структурирование самих файлов проекта или комментирование частей кода для того, чтобы не потеряться в огромном количестве файлов, и чтоб кто-то, кроме самого верстальщика смог потом в этом коде разобраться.

Я сталкивался с подобным только при написании дипломной работы в университете. Это был проект на ASP.NET MVC 5. И, собственно, писал я его в Visual Studio с программными расширениями Identity и Entity Framework. Собственно, сам VS обеспечивает довольно удобное структурирование программных файлов, а Entity, например, самой базы данных. В итоге, если в процессе согласования с преподавателем появлялась необходимость добавить какой-то функционал и т.п., проблем не возникало, так как не затрагивались уже существующие файлы, нужно было просто добавить новый каталог, новую ассоциацию с Entity и писать программные файлы.

\_\_\_\_\_

**Q3:**

При верстке под различные устройства важно учитывать особенности каждого их них. У разных браузеров свои интерпретаторы. Хоть стандарт один, во-первых, не все браузеры поддерживают новейшие свойства/атрибуты и т.д., во-вторых, если и поддерживают, то синтаксис может отличаться.

Чтобы проверить правильность отображения всех элементов можно воспользоваться множеством сервисов, созданных для этого. Например, в Chrome при просмотре кода есть специальный интерфейс, позволяющий выбрать устройство и посмотреть, как с учетом его размеров экрана будет отображаться сайт. А для функций и атрибутов я пользуюсь https://caniuse.com.

\_\_\_\_\_

**Q4:**

Для непосредственно написания кода я использую Sublime или Atom. И в том, и в другом установлен Emmet, который неплохо помогает сократить рутинный набор разными макросами.

А если глобально, то часто использую Bootstrap для разметки. Хотя, иногда и для базового дизайна.

\_\_\_\_\_

**Q5:**

Доступ к исходному всегда можно получить :) в Chrome, например. И посмотреть, какие параметры и атрибуты там используются. Скорее всего, это будет связано с WebKit’ом.

На практике с подобным сталкивался, когда нужно было изменить внешний вид скролла. Тогда возникли проблемы с Safari, как раз. В итоге, я просто загуглил, как правильно указывать эти свойства для этого браузера и добавил в код.

\_\_\_\_\_

**Q6:**

Всегда можно уточнить что-то повторно. Если такой возможности не будет, сделаю на свое усмотрение что-то в стиле: если кнопка с прозрачным фоном, его можно залить при наведении, или подсветить внешней тенью и т.п.

\_\_\_\_\_

**Q7:**

Сейчас я прохожу курсы на Udemy. До этого смотрел видео-уроки на YouTube. Хотя, когда учил в университете C++/C# даже книжки читал, но это в прошлом.

Какую-то часть информации черпаю с разных пабликов ВК или Телеграм. Есть очень классный паблик Библиотека программиста (https://vk.com/proglib). Иногда ворую оттуда что-нибудь, но и лицензию приобретаю тоже, так что совесть чиста :)

Буквально вчера наткнулся на книгу You Don't Know JS. Из интересного: она опубликована на GitHub (https://github.com/getify/You-Dont-Know-JS) и, вроде как, каждый желающий может вносить правки, либо просто читать.

\_\_\_\_\_

**Q8:**

Мне 22 года, я этим летом только закончил университет, в котором занимался программированием под Windows/Linux и т.п. Там же заинтересовался вебом и, собственно, диплом писал под эту сферу.

В сентябре я ушел из компании Mamba (которая для знакомств) и уехал в Израиль, где и сейчас нахожусь. Изучаю все, что так или иначе относится к web-разработке и пытаюсь как можно скорее закончить курс на Udemy, чтобы стать более-менее full stack’ом.

Параллельно ищу различные вакансии, требования к ним.

Предел моих мечтаний: ментор в лице работодателя, под руководством которого я бы смог набраться опыта, еще подучиться, чтобы дальше расти в этой сфере. Сейчас иногда возникают ситуации, когда хочется что-то сверстать, но идей или инструментов не хватает. Приходится ограничиваться заданиями из курса.

Также, хотел отметить, что я слабо знаю JS. Думаю, по моему тестовому заданию это и так станет ясно. В свое оправдание вернусь к тому, что только недавно начал его изучать. Там на моем GitHub есть еще один проект с его использованием. К сожалению, это мой максимум на данный момент.

\_\_\_\_\_

Моя персональная страница: https://evgkuzin.ru

Мой GitHub лэндинг: https://evgkuzin.github.io