**Level I**

Q1

Расскажите, чем, на ваш взгляд, отличается хорошее клиентское приложение от плохого с точки зрения пользователя, менеджера проекта, дизайнера, верстальщика, серверного программиста.

Для пользователя:

* Интуитивно-понятный и приятный интерфейс (юзабилити);
* Быстродействие;
* Приложение должно соответствовать всем ожидаемым требованиям (описанных в документации, инструкции или справке);

Для менеджера проекта:

* Доступность в донесении информации заказчику (например, почему сделана какая-то функциональность именно так);
* Возможность быстро или в кратчайшие сроки добавлять необходимый функционал;

Для дизайнера:

* В каком-то смысле дизайнеры не касаются непосредственно приложения, поэтому для них важно создать мок-ап или макет, который в дальнейшем возможно будет реализовать в коде;

Для верстальщика:

* Макет приложения должен быть прост в реализации;
* Клиент должен быть адаптивным и кроссбраузерным;

Для серверного программиста:

* Использование последних стабильных версий фреймворков и других технологий;
* Обеспечение безопасности через защищенные протоколы и технологии (HTTPS, SSL/TLS);
* Разработанное API должно генерировать валидные ответы и корректно обрабатывать запросы.

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.

* Большой объем написания кода, пакетов, модулей;
* Обычно код пишется командой разработчиков разного уровня знаний;
* Legacy код, т.е. устаревший код, который более не поддерживается и не обновляется, но используется.;
* Использование различного стека технологий, фреймворков в различных частях приложения;

В проектах в которых я участвовал, использовался препроцессор и шаблонизатор Pug, CSS препроцессор Sass, язык PHP на бекэнде. Сайты были развернуты на Bitrix с использованием веб-сервера nginx.

Также в проектах использовал C# для бекэнда, синтаксис разметки Razor, подход MVC/ Вся информация, используемая в приложении, хранилась на сервере базы данных MSSSQL.

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?

Presentational Сomponents - компонент, использующийся исключительно для отображения. Не имеет состояния. Т.к. данные приходят снаружи, его можно без проблем переиспользовать в другой части приложения. По сути является чистой функцией. Сontainer Сomponents - компонент, в которм работают с данными. Имеет свое внутреннее изменяемое состояние.

Q4

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

Классическое наследование - через прототип. В веб-студии, где я работал ранее, делали сайты, и там js был "простой", без классов, наследования.

Q5

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

На данный момент, я знаком только с юнит-тестированием, end-to-end мне предстоит изучить. Так как React изучать начал недавно, я еще не успел применить на практике тестирование. Однако в теории знаю, что тестирование проводится с помощью Jest и Enzyme и я попытался применить этот стек технологий в данном тестовом задании насколько мне позволили знания.

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. Я бы реализовал простой, но опрятный внешне вариант, и показал его коллегам-дизайнерам, вместе обсудили бы, и, либо приняли, либо доработали. В противном случае, следует обратиться к непосредственному руководителю для составления дальнейшего плана действий.

Q7

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

Линтеры, препроцессоры, система сборки. IDE также упрощает процесс написания 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?

Источник актуальной информации Habr. Вечный спутник программистов StackOverflow. Полезные статьи на Tproger. Видео курсы на Youtube. Курсы на Coursehunter. Короткие курсы на hexlet.

В свободное время я могу почитать блоги известных разработчиков или статьи о событиях, происходящих в IT мире. Насчёт областей знаний, не относящихся к работе, люблю погулять с собакой, почитать книги и посмотреть сериалы. И, конечно же, музыка, которая настраивает на рабочее настроение.

Q9

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

<https://arturpravdivtsev.github.io/workout.github.io/>

<https://github.com/ArturPravdivtsev/E-learning-portal>