**Орлова Софья, 8-915-126-85-50**

1. **Расскажите, чем, на ваш взгляд, отличается хорошее клиентское приложение от плохого с точки зрения**

**• пользователя:**

- контент (должно быть интересным, полезным и уникальным)

- удобство использования и возможность быстро найти то, за чем пришел (юзабилити),

- дизайн (должно визуально привлекать, оставлять приятные впечатление),

- безопасность;

**• менеджера проекта:**

- должно выполнять свою задачу (увеличение продаж, привлечение аудитории, автоматизация процессов, получение обратной связи и прочее),

- должно быть легко поддерживаемым, минимум затрат на поддержку и продвижение (понятный, упорядоченный код, наличие тестов, популярные технологии);

**• дизайнера:**

- должно быть интуитивно для пользователя,

- не перегружено излишним функционалом,

- легко воспринимаемо (шрифты, расположение текста, изображаение, подача информации),

- акцентирование на значимых вещах;

**• верстальщика:**

- должно быть адаптивным,

- должно быть кроссбраузерным,

- должно легко проходить переполнение контентом,

- должно быть доступным (работа с клавиатуры, тачпада, чтение сайта голосом);

**• серверного программиста:**

- код должен быть должно быть легко поддерживаемым и оптимизированным,

- предусмотрены варианты реагирования на возможные ошибки,

- быстрота работы приложения,

- наличия резервов при увеличении нагрузки на приложение,

- обеспечение безопасности приложения

1. **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.**

Разработки многостраничного сайта требует детальной проработки его структуры, контента, плана продвижения и поддержки. Я, к сожалению, пока не сталкивалась на деле с разработкой рабочих живых сайтов, но предполагаю, что в отличие от лендинга, многостраничный сайт требует работы большой команды специалистов (дизайнеры, верстальщики, программисты, копирайтеры, маркетологи и прочее), что требует в свою очередь организации и контроля работы всей команды. Крупный сайт требует дополнительных технологий с целью ускорения, упрощения и автоматизации разработки (пропроцессоры, сборщики, фреймворки, тестировщики, использование паттернов и прочее). На обучении работала с sass, gulp, mocha, github. Проблемы были с gulp, тк большое количество плагинов, частые обновления и тяжело найти ошибку, если что-то сбоит. Но помощь приходили старшие товарищи и интернет)

1. **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?**

Presentation Components это модули кода, которые отвечают за отображение данных состояния и обладает возможностью доступом на чтение данных. Сontainer Сomponents агрегирует всю логику по управлению и синхронизации данных.

Плюсы: приложение разделяется на модули, которые четко отвечают своему предназначению, что помогает организовать код. Возможность переиспользования модулей. Помогает сделать код более читаемым для других разработчиков при использовании общепринятых паттернов.

Минусы: сложность навигации по модулям (экспорты), легко потерять связь, неуникальные имена переменных в модулях, подход к разделению все равно подразумевает субъективный подход и требует достаточного вникания в порядок организации кода, требует автоматизацию сборки

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

Позволяет при создании новых объектов унаследовать их свойства и методы от уже существующих. Наследование можно организовать с помощью функций конструкторов и свойства prototype, а так же с помощью классов. В последнем проекте pixel-hunter мы работали по паттерну MVP, в котором все view экранов наследовались от класса abstract-view, который возвращал разметку и на ее базе создавал DOM элемент. Так если у экранов были одинаковые методы, я так же создавала для них отдельный класс и наследовала эти методы для избежания дублирования кода.

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

Jasmin и Mocha как самые популярные. На последнем проекте я тестировала отдельные функции, такие как ресайз изображений, запуск таймера, расчет баллов при прохождении игры. Работала с mocha + chai по bdd. Наверное, сложность вызывает, что именно нужно тестировать

1. **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?**

В первую очередь уточню у дизайнера/менеджера проекта наличие макетов по динамике или хотя бы описания. В случае, если по каким-то причинам получить обратную связь не могу, то продумаю возможные действия с данной формой, пропишу примитивные стили по динамике и включу в программу возможные действия. Уточню у серверного программиста, в каком формате и какие данные я должна отправить. Когда получу макеты, подкорректирую css, исключу лишние действия/обработчики. В случае, если не получу макеты вовсе, по согласованию отрисую все на свой тонкий эстетический вкус :)

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

На последнем проекте выручал gulp при сборке и его watcher при изменение кода с динамическим обновлением страницы. Github для контроля версий. Ну и конечно отладчик в браузере

1. **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?**

Learn.javascript, habr, форумы с обсуждениями по подходам решения тех или иных задач, документация mdn - наверно, самые часто посещаемые мной на текущий момент сайты. Но так как я только закончила обучение, то основные материалы к изучению я получала от преподавателей и от наставника.

Я имею опыт работы в финансовых команиях со стороны бизнеса (см резюме) по автоматизации/оптимизации бизнес-процессов, поэтому участие в проектах мой второй большой бэкграунд. Так же интересуюсь психологией - но тут, хочешь - не хочешь, т.к. вся семья у меня психотерапевты :)

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

<https://github.com/SonyaOrlova>

Я люблю изучать новое, общительная, усидчивая и вяжу крючком игрушки :)

