Level I

Q1

Хорошее клиентское приложение отличается от плохого, с точки зрения:

клиента, в первую очередь тем, насколько быстро и легко оно способно решать поставленные цели: быть интуитивно понятным, легко устанавливаться и поддерживаться, оправдывать свое назначение.

менеджера проекта, тем насколько эффективно приложение справляется с поставленными задачами. Например, развлекательный ресурс будет считаться хорошим, если посетители проводят n-ное количество времени, проявляют желаемую активность и т.д., а сам проект сделан в заданные сроки, с минимальным количеством багов и финансовых затрат.

дизайнера, сразу по нескольким критериям. Для хорошего приложения важна как удобность использования, так желаемое поведение публики. Например, известно, что от того, как расположена картинка рядом с кнопкой оплаты на сайте, будет зависеть к-во кликов по этой самой кнопке. Это конечно самый яркий пример, эффективного использования дизайна для поставленных целей, но также на поведение ЦА будет влиять и множество других факторов, включая контент, форму, цвета и т.д.

верстальщика, тем насколько точно следует макету само приложение, и учитывает его ошибки. Например, часто бывают случаи, когда одни и те же элементы могут иметь разные отступы на разных страницах макета, или не будут отображены события при клике на те, или иные кнопки. Поэтому, конечно, здесь очень важна слаженная работа верстальщика и дизайнера.
Также важно чтобы само приложение соответствовало всем современным требованиям: было легко читаемым и масштабируемым, было доступно слабовидящим и слабослышащим людям, соблюдены семантические особенности для выдачи в поисковиках и т.д.

серверного программиста, тем насколько быстро и эффективно проходит обмен данными, насколько защищены от внешних угроз данные пользователя и сам сайт.

Q2

Опыта в разработке многостраничных сайтов, на данный момент, у меня нет, но уверен, что здесь важна масштабируемость и удобство работы с кодом. Здесь хорошо подходит методология БЭМ, а также понятные комментарии от самих разработчиков.

Ранее я разрабатывал двухстраничный сайт, используя функции-конструкторы и классы, с двумя HTML файлами: https://github.com/OlegPeunov/news-explorer-frontend-Peunov

Сейчас я бы уже сделал этот же сайт на компонентах Реакт, подобно другому моему проекту: https://github.com/OlegPeunov/image-uploader/tree/main/src/components

Реакт удобен не только тем, что компоненты можно легко добавлять и убирать, но и тем, что всю необходимую информацию можно передавать через поднятие стейта, и подписку на контекст.

Обе эти технологии я использовал в проекте выше.

Q3

Компоненты **Container** и **Presentational** используются для распределения ответственности, таким образом, чтобы компоненты Conatainer принимали и работали с данными, а компненты Presentational отвечали за отображение этих данных. Благодаря этому подходу компоненты Presentational можно переиспользовать в других частях кода.

Q4

Наследование — это передача данных и функциональности от одного объекта к другому. Удобно использовать, например, в случаях, когда у разных пользователей сайта, есть разные права доступа, и чтобы не плодить сущности можно воспользоваться наследованием классов, создав конструктор, который унаследует свойства и методы у другого конструктора. Лично мне не приходилось пользоваться наследованием в похожих проектах, но смогу легко применить его при поддержке существующих проектов Funbox.

Q5

Ранее, я изучал работу с библиотекой Jest, и настраивал несколько тестов в рамках курса по бэкенду от Яндекс.Практикум. В реальных проектах мне не приходилось интегрировать эту библиотеку, но буду рад такой возможности. Уверен, что смогу также освоить и рбаоту прочих библиотек, используемых в ваших проектах.

Q6

В случае отсутствия подробных инструкций по работе формы, я бы составил список функций, которые я считаю необходимыми, и отправил на согласование руководителю. Список мог бы состоять из следующих пунктов:

1. Условия открытия и закрытия формы.
2. Проверка корректности ввода данных. Например, можно настроить автоматическую проверку или написать регулярные выражения.
3. Вывод сообщений, предупреждающих о той или иной ошибке ввода данных. Также я бы подробно прописал, какие именно сообщения будут всплывать при тех или иных событиях.
4. Активация и дезактивация кнопки субмита, в зависимости корректности ввода данных.

Q7

Для отладки кода я пользуюсь дебаггером, панелью инструментов разработчика в браузере. Для эффективной работы и тайммменджемнта, мне также очень помогает разбитие проекта на задачи, которые я тщательно прописываю в доске на сайте Trello. Это позволяет работать эффективнее и не пропускать дедлайны.

Q8

Для профессионального развития я прохожу курсы, например на Степик и Яндекс.Пракиткум, смотрю популярные ютуб каналы, например, Владлена Минина или IT-KAMASUTRA.
В данный момент читаю профильную литературу: «Выразительный JavaScript» и «React и Redux. Функциональная веб-разработка».

Помимо веб-разработки меня интересуют иностранные языки, орнитология, поп-сайенс.

Q9

Меня зовут Олег Пеунов, я начинающий front-end разработчик.

Недавно я закончил курсы web-разработки на Яндекс.Пракиткуме и теперь продолжаю самостоятельно совершенствовать свои навыки в JS и React.
Я владею навыками JavaScript, умею верстать и работать с DOM. В данный момент работаю с React, а годом ранее проходил обучение по бэкенду, поэтому имею работать и с Node.js. Также я имею опыт работы с webpack и nginx.
Помимо этого, у меня есть достаточный для работы уровень английского, т.к. для меня это был единственный язык общения на одном из международных проектов, в котором я учувствовал на протяжении года.
Уверен, что смогу быстро научиться всему необходимому и освоить работу с Docker, k8s и автоматизированными тестами.

Среди выполненных работ могу отметить сайт для поиска новостей. В проекте применены такие технологии как: JS, HTML, CSS, Git, NodeJS, WebPack
Сайт используется для поиска новостей по запросу, с возможностью хранения новостных карточек в личном кабинете: https://github.com/OlegPeunov/news-explorer-frontend-Peunov

Бэкенд для хранения данных пользователя, создан с применением следующих технологий: JS, Git, Node, Mongo, Яндкс-Облако.
Бэкенд для работы с данными пользователей, полученных с news-explorer-api-Peunov.
Подробнее: https://github.com/OlegPeunov/news-explorer-api-Peunov

Сайт для хранения изображений, создан с применением технологий: React, JS, Git, Node, VS Code
На сайте можно просматривать, добавлять и удалять свои собственные изображения, а также менять данные пользователя.
Подробнее: https://github.com/OlegPeunov/image-uploader