**Квалификационные задания для разработчиков JavaScript**

**Level 1**

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

**• пользователя;**

**• менеджера проекта;**

**• дизайнера;**

**• верстальщика;**

**• серверного программиста.**

Пользователь: быстродействие, юзабельность, дизайн

Менеджера проекта: бюджет проекта, планирование

Дизайнер: UX/UI дизайн, юзабилити

Верстальщик: адаптивность, кроссбраузерность, валидность

Серверного программист: оптимизация, быстродействие, архитектура приложения, логика, правильная передача данных и их обработка.

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.**

Опыт:

Мой друг попросил сделать ему интернет-магазин на тематику чехлов. Макеты нарисовал он, я их сверстал адаптивно, кроссбраузерно и валидно. Пожалуй, главная страница была одна из сложных страниц, так как там было около 7 каруселей, разной сложности реализации.

Был выбор между Wordpress и OpenCart. Я выбрал Wordpress так у него больше бесплатных плагинов, тем, информации. Для реализации интернет-магазина использовал плагин Woocommerce. Сложности вызвало натяжка на Woocommerce, но покопавшись в официальной документации, зарубежных форумах я находил что хотел. Знаний по PHP мне хватило для написания небольшого плагина для вывода категорий, подкатегорий.

Были проблемы с семейством шрифтом Bebas, так как он не корректно отображался на странице, решил эту проблему через генератор шрифтов.

Также были проблемы с кроссбраузерностью когда некоторые браузеры некорректно отображают контент, решил я ее использованием префиксов в CSS.

Для сборки проекта использовал Gulp. Показывал готовую верстку на GitHub.

В этом проекте использовал: Html, CSS, JavaScript, jQuery, PHP, MySQL, Wordpress, Woocommerce.

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?**

Я поддерживаю данную архитектуру разделения на Presentational Сomponents и Сontainer Сomponents.

Сontainer Сomponents : я создавал компонент высшего порядка, который передавал контекст компонентам и это очень удобно.

Плюсы:

1. Код легче читается
2. Логику Сontainer Сomponents можна использовать многоразово для многих компонентов.
3. Presentational Сomponents позволяют работать c HTML, CSS не затрагивая логику компонента.

Минусы:

Пока не заметил

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

**Наследование** — один из четырёх важнейших механизмов объектно-ориентированного программирования.

Наследование позволяет описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

Наследование в Js реализуется с помощью оператора “extends”:

Class <child> **extends** <parent> {….}

В ES5 наследование реализуется с помощью вызова родительской функции в дочерней функции :

Parent.**call**(props) или Parent.**apply**(props, …..)

Опыт:

Я пытался написать свой конструктор сайтов. Там были разные типы полей(цвет, select, radio, текст…). У всех них есть много общего: значение, селектор, события при изменении… Я создал родительский класс который связал все общие свойства и методы. Потом наследовал к разным типом полям.

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

Функции тестирую с помощью: Jest, chai, mocha

Компоненты в React тестирую: Enzyme

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?**

Многое зависит от сути формы. При отправке формы буду проверять наличие и корректность данных из формы на JavaScript. Отправляться форма будет через Ajax. В случае если поле пустое или некорректное выводить сообщение об этом пользователю. Также нужно проверять данные на сервере. Если сервер одобрил все данные , то выводит сообщение об успешной отправке.

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

В написании: Gulp,Webpack, Emmet, Sass, Avocode, Bootstrap, jQuery

В проверке: DevTools, Jest, Enzyme, chai, mocha

Отладка кода: DevTools

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?

**Ресурсы:**

1. <https://learn.javascript.ru/>
2. <https://developer.mozilla.org/ru/>
3. <https://habr.com/>
4. <https://medium.com/>

**Форумы:**

1. <https://stackoverflow.com/>
2. <https://toster.ru/>
3. <http://qaru.site/>

**Видео:**

1. Glo Academy <https://www.youtube.com/channel/UCVswRUcKC-M35RzgPRv8qUg>
2. WayUp <https://www.youtube.com/channel/UCrWP9UeNep74kW0cxY7YPpw>
3. WebDesignMaster <https://www.youtube.com/user/agragregra>
4. Быть программистом <https://www.youtube.com/user/CRIK0VA>
5. Max Frontend <https://www.youtube.com/channel/UCqJyAVWwIqPWKEkfCSP1y4Q>

Помимо веб-разработки изучаю физику, природные явления, животных.

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

Я Пужаков Игорь. Родился в 2001 году в городе Белгороде-Днестровском. Занялся веб-разработкой еще 2 года назад. Началось все неудачно, так как я самоучка (начал изучать PHP, так как где-то услышал).

Потом изучил Html, CSS. Далее начал изучать JavaScript, и он мне больше понравился чем PHP. Спустя некоторое время взял заказ на интернет-магазин. Выполнил за несколько месяцев, так как знаний в разработке интернет-магазина было на нуле. После этого я хотел попрактиковаться и написать небольшой конструктор сайтов. И понял что для лучшей оптимизации, скорости загрузки, удобности нужно использовать фреймворк. Выбрал React, так как разделения на компоненты мне очень понравилось. Чуть позже начал использовать связку React+Redux. Написал несколько приложений.

Сложные и неизвестные задачи больше всего меня привлекают, так как благодаря им я каждый раз совершенствуюсь. Занимаюсь спортом : бег, настольный теннис, калистетика. Читаю книги по тематикам: детективы, романы, психология, личный рост.

Последние работы:

<https://gonor.com.ua/>

<https://kaigor619.github.io/gambda/>

<https://kaigor619.github.io/lian/>

<https://kaigor619.github.io/qenrique/>

<https://kaigor619.github.io/colorpicker/>

<https://kaigor619.github.io/re-store/>

<https://kaigor619.github.io/todo/>

**Level 2**

Репозиторий : <https://github.com/kaigor619/testapp>

Результат: <https://kaigor619.github.io/yandex-map/>