

**Q1.** Расскажите, чем, на ваш взгляд, отличается хорошая верстка от плохой с точки зрения: пользователя, менеджера проекта, дизайнера, верстальщика, клиентского программиста, серверного программиста.

В первую очередь хорошая верстка отличается от плохой ....

- правильным отображением на любых устройствах/платформах,
- соответствием макету от дизайнера,
- скоростью загрузки и вниманием к performance-тестам: помним про ограничение скорости интернета, максимально все минифицируем, разделяем стили media-запросами, и т. д.
- масштабируемостью функционала (прикручиваем новые штуки без головной боли: все текущие элементы верстки не страдают от введения новых),
- понятным названием классов и id, а также их переопределением и расширением,
- вниманием к психологии человека: максимально быстро даем пользователю отклик, а неконтролируемые ситуации зависящие от third-party сервисов закрываем прелоадерами и няшными анимациями
- адаптивностью и responsive-функциональностью (идет последним пунктом, т. к. зависит от задачи, иногда это overkill-функциональность)

Пользователь	Сайт/Приложение работает так как должно, пользователь счастлив. Он пришел с какой-то целью и получил результат без головной боли :)
Менеджер проекта	Нововведения и правки вводятся максимально быстро и эффективно, соответствуя требуемому результату.
Дизайнер	Сверстаный сайт в точности соответствует макету
Верстальщик	Сайт/Приложение правильно отображается на любых устройствах и платформах, введение новой функциональности не вредит уже существующей верстке. Минимизирована повторяемость одних и тех же реализаций стилей, атрибутов и блоков кода.
Клиентский программист	Каждому блоку и элементу верстки даны осознанные названия, а значит клиентскому программисту не приходится чрезмерно вникать не в его область. Вся работа со стилями (на сколько это возможно) выполнена на этапе верстки, чтобы сейчас уже решались вопросы динамического изменения страницы и контента.
Серверный программист	Понятная файловая структура и разделение кода

**Q2.** Опишите основные особенности верстки крупных многостраничных сайтов, дизайн которых может меняться в процессе реализации и поддержки.

“Мухи отдельно, котлеты отдельно”. Важно разделить макет на одиночные сущности которые будут располагаться на странице и для каждой отдельно описать стили.

Создать несколько стартовых шаблонов страниц для разного типа контента (например, для новостного сайта — лента, страница новости, блок комментариев и т. д.; для личного кабинета интернет-провайдера — главная страница с балансом, страница настроек, лента обращений в поддержку).

Получаем понятную структуру элементов, с которой можно работать не вникая в полную верстку страницы, и ре-использовать уже созданные элементы в других частях проекта, немного подправляя исходную верстку.

В целом это похоже на ООП: абстрагируем элементы, чтобы каждый из них был законченным, и инкапсулируем их так, чтобы они были универсальны в любой части проекта, разделяем проект на блоки чтобы легче было поддерживать.

Расскажите о своем опыте верстки подобных сайтов: какие методологии, инструменты и технологии вы применяли на практике.

Я делал верстку для SPA-приложения со стилями на Bootstrap, поэтому каждая страница сразу же начиналась с заранее заготовленного шаблона, где в теге `<body>` уже лежал контейнер для адаптивной верстки. Это начало каждой страницы.

Информация размещалась в виде карточек в двух вариантах: либо в виде доски (по аналогии с Trello), либо в виде горизонтального списка с прокруткой.

Единицей сущностью контента там выступала карточка: заранее был подготовлен ее шаблон с полями под заголовок, отступами и базовым стилем. Сразу же были предусмотрены дополнительные стили для разных вариантов отображения: с цветовым выделением, тенями и т. д.

То есть изначально определяем ширину контейнера, размеры шрифтов и типографику, точки перехода к другому layout'у карточек для мобильной версии, затем варианты расположения карточек на странице, а уж потом базовые варианты оформления карточек. Это основа.

Теперь отталкиваясь от этого набора можно сверстать любую типовую страницу, где тело карточек будет отличаться в зависимости от типа контента, который в них размещается.

**Q3.** Опишите основные особенности верстки сайтов, которые должны одинаково хорошо отображаться как на любом современном компьютере, так и на смартфонах и планшетах под управлением iOS и Android.

Тут 2 проблемы: адаптивная верстка и поддержка различных браузеров.

Первое решается использованием какой-нибудь библиотекой вроде Bootstrap, которая является стартовой точкой адаптивной верстки с уже заданными размерами контейнеров под разные размеры экранов.

Затем медиа-запросами задаем брейкпойнты для изменения вида страницы в зависимости от размера экрана. В крайнем случае мобильная версия выносится в отдельное место под другим URL.

Читаем документацию для разных мобильных ОС таких как iOS и Android и прописываем необходимые мета-теги для правильного отображения страниц на мобильных устройствах.

Например, на iOS обыкновенный сайт можно замаскировать под веб-приложение, добавив короткое название, иконку на Springboard, экран загрузки, а также описать правила выреза под челку для современных моделей iPhone.

Поддержка различных браузеров решается использованием альтернативных описаний стилей в некоторых ситуациях. Например иногда для WebKit надо добавить пару дополнительных тегов для плавной анимации. Дополнительно каждый раз когда используем что-то вроде flexbox или новее, лучше сходить на [caniuse.com](http://caniuse.com) и дополнительно проверить совместимость со старыми браузерами.

Расскажите о своем опыте верстки подобных сайтов: какие инструменты и технологии вы применяли, как проверяли результат на различных устройствах, какие именно устройства требовалось поддерживать.

Я начинал разрабатывать свое SPA-приложение с расписанием для ВУЗа, использовал Bootstrap и Bootstrap Studio для верстки.

Как правило смотрю на адекватное отображение сайтов в Safari и Chrome: это покрывает стандартные браузеры двух основных мобильных ОС — iOS и Android, а также все настольные браузеры вроде Chrome, Яндекс.Браузера, Edge, а также Safari на macOS.

Обычно проверяю адаптивность верстки с помощью встроенных инструментов разработчика в Chrome, там же смотрю на вкладке Performance скорость и порядок загрузки элементов, ну и Waterfall. Если необходимо — всякие онлайн-тесты для проверки скорости и выставления оценок быстродействия.

Дополнительно все тестирую с личных устройств которые есть под рукой.

**Q4.** Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.

Sublime Text, Atom, Bootstrap Studio, инструменты разработчика в Chrome, React DevTools. Когда при отладке работаешь с веб-приложениями — React и Webpack также “ругаются”: один в консоль для отладки JS, другой — в терминал где развернута Нода. Ну и наконец Google и StackOverflow :)

**Q5.** Вам нужно понять, почему страница отображается некорректно в Safari на iOS и в IE на Windows. Код писали не вы, доступа к исходникам у вас нет. Ваши действия? Сталкивались ли вы с подобными проблемами на практике?

Опять же в первую очередь полезут инструменты разработчика Chrome, скорее всего дело с поддержкой каких-то редких CSS-атрибутов WebKit’ом (если проблема на iOS) или вовсе какой-то файл не загрузился. Прогоню онлайн-утилиты на совместимость сайта с IE. Достану с полки старый Windows-ноут, открою DevTools в IE и буду смотреть что рендерится не так, как должно.

С подобными проблемами сталкивался, но не часто.

**Q6.** Дизайнер отдал вам макет, в котором не показано, как должны выглядеть интерактивные элементы при наведении мыши. Ваши действия?

Тут же напишу в чатик про эту проблему, параллельно от себя добавив простые стили вроде затемнения кнопок при наведении мыши.

Возьму стандартные стили Bootstrap’а и посмотрю насколько там затемняются элементы при наведении мыши. Если это смотрится органично — пропишу в стилях похожее затемнение, пока в чатике тишина.

**Q7.** Какие ресурсы вы используете для развития в профессиональной сфере? Приведите несколько конкретных примеров (сайты, блоги и так далее).

Coursera, Хабр, Вастрик, Типичный программист. Часто читаю блоги на Medium про организацию движков крупных сайтов вроде “Монитора” и “Антихайпа” от Медузы или “Основы” от Комитета. Если что-то супер-срочно — храни ребят из Индии с их видеоблогами на YouTube.

Ну и гуглить how to [вставить навык] with React очень спасает :)

Какое направление развития вам более близко: JS-программирование, HTML/CSS- верстка или и то, и другое?

В общем-то мне нравится весь frontend и все что с ним связано: я люблю делать макеты в Sketch (правда больше мобильные приложения), заниматься HTML/CSS версткой, ну и наконец React.

Какие ещё области знаний, кроме тех, что непосредственно относятся к работе, вам интересны?

За время работы в пресс-службе ВУЗа в задачи входило:

- правки контента на страницах, разработка идей и их добавление на сайт (работая вместе с ребятами из отдела разработки)
- идеи по дизайну и предложения по увеличению целевых действий на страницах за счет изменений в макет,
- разработка своих макетов (кусочки веб-страниц, email-рассылки, печатная продукция и т. д.),
- разработка стилей для постов, оформление соц. сетей, согласование баннеров, и т. д.
- реклама, таргет, креативы, и прочие штуки,
- сбор статистики и работа с метрикой.

Помимо этого я занимаюсь фото- и видео- продакшеном, делаю репортажи, промо-ролики, интервью, медиа-киты для соц. сетей, спец. проекты и т. д.

Из софта “не по специальности” умею в Sketch, Photoshop, Pixelmator, Lightroom, Final Cut Pro X.

Ну и наконец — работаю со сценическим светом и организую чуть ли не самые замороченные домашние вечеринки в Москве: дома всего лишь есть 20+ удлинителей, около 40 ламп, дым-машина и дискошар диаметром больше полуметра. Часть из всего этого хаоса работает с помощью самодельной системы управления на macOS через Arduino.

Иногда вот так гремим до 5 утра, а я при этом играю на виниле :)

**Q8.** Расскажите нам о себе и предоставьте несколько ссылок на последние работы, выполненные вами.

Меня зовут Олег, мне 20 лет, учусь на 4 курсе в МАИ на IT-специальности.

Интересуюсь разработкой, брался за Java, Swift, но вот веб и React больше всего зацепил. Обожаю хороший UI/UX и типографику: пресс-служба универа и отдел разработки Ozon’a прислушиваются и даже просят больше показывать их проблемы)

Я многого могу не знать, но именно поэтому хочу к вам на вакансию Junior — поработать над реальными проектами в команде, где подскажут если я что-то делаю не так. В то же время я могу сверстать и закодить практически что угодно, потому что любой вопрос можно нагуглить и разобраться на практике.

Обожаю все делать максимально качественно и вылезать за рамки только своей зоны ответственности: идей обычно много, вдруг что приживется или поможет проекту стать лучше.

Из последних работ в рамках учебы в универе делал SPA-приложение на стеке MERN (да, это очень похоже на ToDo-менеджер с каких-нибудь курсов, но нас из-за карантина просто “бросили в воду”, так что пришлось самостоятельно вникать):

— репозиторий: [https://github.com/okuznetcov/movies-app\\_mai](https://github.com/okuznetcov/movies-app_mai)

— описание: [https://github.com/okuznetcov/movies-app\\_mai/raw/master/report%26summary.pdf](https://github.com/okuznetcov/movies-app_mai/raw/master/report%26summary.pdf)

Начинал делать расписание для универа с элементами таск-менеджера и соц. сети, на данный момент на этапе верстки:

<https://github.com/okuznetcov/smenaio>

В рамках работы в пресс-службе занимался сайтами “Траектория Взлета” и “Предуниверсарий МАИ”, но больше участвовал в наполнении, ТЗ разработчикам и правках, чем самостоятельной разработкой:

<https://traektoria.mai.ru/>

<https://preduniversariy.mai.ru/>

Level2 тестового задания находится в репозитории здесь:

[https://github.com/okuznetcov/funbox\\_testTask](https://github.com/okuznetcov/funbox_testTask)