1. В качестве сборщика проектов используется Gulp. Gulp настроен таким образом, чтобы максимально увеличить скорость разработки проекта.
2. В качестве HTML - шаблонизатора используется Pug с методологией БЭМ, а также семантика, именование страниц с умом (“button” - для кнопок, “a” - для ссылок, “h..n - для заголовков” и т.д.) . Существует единый шаблон Pug, который изменяется под структуру каждого проекта индивидуально. Во время написания кода необходимо максимально использовать возможности шаблонизатора. Результатом компиляции являются HTML - страницы содержащие структурированную, понятную разметку с комментариями, если это необходимо.
3. CSS препроцессор – Stylus. После компиляции генерируется файл \*.min.css и \*.css
4. Верстка осуществляется с использованием единиц измерения em, rem, px
5. JS - код пишется структурированный, понятный, нативный (без jQuery). Наименование функции, методов, объектов - в соответствии с выполняемыми ими действиями. После компиляции генерируется 3 js-файла:

* lib.js - содержит сконкатенированный код всех подключенных библиотек;
* \*.js - пользовательский код;
* \*.min.js - минимизированная версия пользовательского кода.

1. Изображения подготавливаются в формате progressive JPG, PNG (должны быть уменьшены без потери качества). SVG файлы генерируются в svg-sprite, png-иконки - в icon-sprite.
2. При реализации форм, предусмотреть валидацию полей;
3. При наличии в проекте модальных окон, таковые размещаются на всех страницах проекта с возможностью их вызова, где это необходимо. Перед каждым модальным окном есть комментарии, описывающий его назначение и место использования.
4. Если HTML элемент содержит в себе какие-либо js-событие, то этот элемент именуется отдельным классом, по которому и будет инициализироваться нужно событие. Важно, чтобы данный класс не содержал стилевых свойств.
5. Если при инициализации библиотеки или кастомной функции какой-либо параметр предположительно будет изменен, данный параметр следует вынести и передавать его параметром в функцию.
6. Каждый сверстанный блок, выносится в отдельный .pug && .styl file && image directory, далее подключается в общем шаблоне styles.styl (для стилей), index.pug (для шаблонов), директории с картинками именуются по названию блока.

Пример “блок рекламной информации” должен находится в папке b-information, которая содержит: b-information (директория для изображений), b-information.styl (файл стилей данного блока, в нем же описывайте медиа запросы, которые касаются только данного блока), b-information.pug (файл шаблона).

1. Наименование спрайтов: в папке с изображениями отдельного блока необходимо добавить директорию “sptites”.
2. Для реализации разметки используйте flexbox, работы с подключенными фреймворками не принимаются, так же не забывайте об autoprefixer.

Медиа-запросы css. Используйте контрольные точки (1024, 768) количество точек можно увеличить, при необходимости можно добавить landscape ориентацию страницы.

Должно быть корректное отображение всей информации согласно ТЗ, проверяться ТЗ обязательно будет на популярных телефонах, планшетах, а также при изменении размеров окна браузера на мониторе.

1. Блоки разрабатываться кроссбраузерно и адаптивно, а также должны быть ретинорезированными (используется только векторная графика (в случаях с размытым фоном, используется растровая)), минимальные требование к браузерам:

* IE >= 10
* Safari >= actual
* Mobile Safari (iOS >=9)
* Mobile Chrome (iOS >= 9)

1. Работы принимаются со ссылкой на репозиторий (например: github, bitbucket)