Базовые требования

1. **Кроссплатформенность:** Windows, MacOs, планшетники iOs, планшетники Android
2. **Кроссбраузерность:** IE7+, Chrome (15+), Firefox (15+), Opera (9+), Safari 4+ – верстка полностью соответствует дизайну, скрипты работают в соответствии с ТЗ. В IE могут быть незначительные упрощения, скрипты, связанные с анимацией, могут работать в упрощенном виде.
3. **Стандарты:** HTML5/CSS3. HTML должен проходить валидацию.
4. **Соответствие макету:** при накладке допустимы незначительные различия.
5. **Ширина сайта:** варьируется от 500 до 1280, сайт выровнен по центру. При 640+ должна отсутствовать горизонтальная прокрутка.
6. **Подвал прижат книзу.**
7. **Расширяемость:** блоки с изменяющимся содержимым подстраиваются в соответствии с дизайном при уменьшении/увеличении контента. Иными словами, все блоки, высоту которых ничего не мешает в дизайне сделать динамической, должны иметь именно динамическую (т. е. зависимую от содержания) высоту, а иногда, чтобы ничего не могло потенциально поломать дизайн, нужно задавать и минимальную высоту. Если хотите сделать блок фиксированной высоты — сначала спросите у заказчика. Для текстовых блоков, заполняемых с помощью визуальных редакторов, должны быть прописаны стили оформления всех тегов контента.
8. **Масштабирование страниц:** страницы при масштабировании в диапазоне 70-150% в браузерах IE9+, Chrome 15+, Opera 12+, Safari 5+, FF15+ должны выглядеть так же, как и при 100%. Допускаются небольшие погрешности, которые возникают из-за неправильных округлений координат браузеров. Масштаб в IE7-8 не проверяется.
9. **SEO оптимизация**: использование микроданных (microdata). Поднять контент как можно выше по коду.

HTML код

1. **Кодировка**: utf-8
2. Структурный, комментируемый код (обозначается начало/конец крупных блоков). Отбивка табами.

<!--—BEGIN FOOTER -->

<!--—END FOOTER -->

1. Семантическая разметка на уровне грамотного использования тегов.
2. Для всех элементов, которые могут содержать текст различной длины, который должен быть вписан в одну строку (например, для кнопок или заголовков, если в дизайне не предусмотрено, что они могут занимать больше одной строки), обязательно задавайте CSS свойство white-space:nowrap.

CSS код

1. Структурный, отбивка табами. Пример:

/\* \_\_\_\_\_\_\_\_\_\_\_1. Сброс CSS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

/\* \_\_\_\_\_\_\_\_\_\_\_2. Типовые элементы\_\_\_\_\_\_\_\_\_\_\_\_\*/

/\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.1. Залоговки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

/\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.2. Ссылки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

/\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.3. Элементы форм\_\_\_\_\_\_\_\_\_\*/

/\*\_\_\_\_\_\_\_\_\_\_\_3. HEADER (Шапка сайта) \_\_\_\_\_\_\_\_\_\_\*/

/\*\_\_\_\_\_\_\_\_\_\_\_4. FOOTER (Подвал )\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

/\*\_\_\_\_\_\_\_\_\_\_\_5. SIDEBAR (Справа)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

1. Комментариями обозначены начало/конец крупных модулей/блоков разметки.
2. Названия классов и id должны по смыслу соответствовать применению, например: header, menu, footer, news
3. Допускается использование вендорных префиксов.
4. Стили для IE7-8 вынесены в отдельные CSS.
5. Для IE7-8 для реализации неподдерживаемых CSS свойств допустимо использование Javascript и expression.
6. Если используете CSS хаки, комментируйте, что это и для какого браузера, а лучше используйте css\_browser\_selector.js. Заботьтесь о верстальщиках, которым придется работать с этим макетом после вас.
7. Всегда описывайте цвет фона для body даже если он белый.

Javascript код

1. Все что можно сделать без Javascript, делать без него.
2. Структурный, отбивка табами.
3. Имена переменных осмысленные, на наше усмотрение.
4. Снабжен комментариями: описаны назначения методов/классов/функций/условий.
5. Для объемных задач используется ООП, для простых – обычные функции.
6. Код должен быть без ошибок.
7. При использовании Ajax и не предоставления заказчиком api для взаимодействия с серверной частью, api делаем на свое усмотрение.

Изображения

1. Имена файлов осмысленные.
2. Проходят базовую оптимизацию на уровне оптимизации для Веб в Photoshop.
3. Малые изображения оформления объединенные по назначению и склеиваются в спрайты.
4. Не пренебрегать возможностью использовать GIF/PNG с 8-битным альфаканалом вместо PNG-24, где это возможно.

Файловая организация:

1. HTML файлы лежат в корне. Главная страница – index.html. Остальные – по именам макетов.
2. Стили в папке /css/
3. Javascript - /js/
4. Изображения оформления - /img/
5. Изображения содержания - /pic/