2. **внешний вид**

УРОВЕНЬ РАСПОЛОЖЕНИЯ СТРАНИЦ

Далее мы также должны знать, что страницы сайта могут быть различной степени вложенности. Различие уровня определяется степенью важности. Как раздел и подраздел.

Например: **http://kaksdelatsite.ru/nemnogo-o-dizayne** это адрес страницы на которой вы находитесь. Где **kaksdelatsite.ru** это адрес сайта как такового, а **nemnogo-o-dizayne** адрес страницы первого уровня, может быть и третьего, четвертого и т.д. уровня страницы. С тем, сколько может быть всего уровней я лично не знаком, но пока с пределом вложенности страниц на сталкивался.

Немаловажно понимать, что чем глубже страница, тем она менее значима для поисковых роботов.

Теперь рассмотрим непосредственно одну из структур сайта которую я использую в своем проекте.

В проекте используется Линейная структура сайта:

Линейная структура сайта Название этой структуры ресурса говорит само за себя: вся информация на сайте предоставляется в линейном виде. Это своеобразная цепочка, состоящая из связанных друг с другом компонентов. Здесь не выделяются разделы, у пользователя нет возможности перемещаться с одной страницы на другую, как ему заблагорассудится. Он может лишь переходить по конкретно заданному маршруту. Поэтому большое значение имеют логическая структура сайта, наличие названия каждой страницы и ссылки на главную страницу. Также желательно указать количество страниц в целом и обозначить ту, на которой в это время находится посетитель. Сфера применения сайтов с подобным размещением страниц весьма ограничена. Структура сайта визитки. Вот яркий пример линейной структуры сайта. Также сайт такой структуры замечательно бы подошел для создания онлайн-книг, к тому же его очень легко реализовать. Виды: разновидностью выше рассмотренной структуры сайта является линейная структура ресурса с ответвлениями. Она подобна дороге с большим количеством ответвлений от нее. Пользователь перемещается с одной страницы на другую в строго определенном порядке. Однако при необходимости он всегда сможет перейти на другую ветвь и без усилий вернуться назад. Главным достоинством линейной структуры с ответвлениями является относительно несложная возможность веб-мастерам создать ее на основе обычной линейной структуры. По мере раскрутки сайта в этом довольно часто возникает потребность. Контент сильно разрастается и возникает вопрос улучшения навигации. Поэтому важен такой процесс, как нарисовать структуру сайта. линейная структура с альтернативными вариантами – еще один подвид линейной структуры сайта. Он отличается от линейной тем, что пользователь имеет большее количество возможностей для поиска информации, а вернее – предоставляется выбор между двумя ответвлениями. Например, разделение на сайте корпоративных и частных клиентов. Чаще всего такую структуру применяют для регистрации посетителей вебсайта. В таком случае все пользователи начинают работу со стартовой страницы. Но потом происходит разделение - частным лицам предлагается одна форма для заполнения информации, а представители коммерческих структур заполняют совсем иную форму.

Источник: http://vistanews.ru/computers/internet/68932 ©

Так же на сайте есть опрос состоящий из PHP кода

**Например**:

*if (file\_exists("$id.dat")) {*

*$data = file("$id.dat");*

*if ($vote) {*

*foreach($ip\_abbr as $value)*

*if ($ip == $value) {echo "<p><b><font color=red> Вы уже голосовали! </font></b></p>";*

*exit;*

*}*

Этот код

1. Проверяет существует ли это голосование
2. Читает содержимое файла результатов и помещает в массив
3. Сравнивает ip с уже записанными ip адресами, если они записаны то повторное голосование запрещено.

Через тег <iframe> вставлен PHP с оформлением HTML в который так же через тег <iframe> вставлен сам PHP опрос.

С помощью этого кода

*$id = (int) $\_GET[id];*

*$vote = (int) $\_GET[vote];*

PHP сможет:

* Получить файлы по определенному голосованию
* Привести к целому числу идентификатор голосования
* Привести к целому числу передаваемое значение голосования

Так же PHP читает содержимое файла результатов и помещает в массив

*$data = file("$id.dat");*

В этом коде

*if ($vote) {*

*$f = fopen("$id.dat","w");*

*flock($f,LOCK\_EX);*

*fputs($f, "$data[0]");*

*for ($i=1;$i<count($data);$i++) {*

*$votes = explode("~", $data[$i]);*

*if ($i==$vote) $votes[0]++;*

*fputs($f,"$votes[0]~$votes[1]");*

*fflush($f);*

*flock($f,LOCK\_UN);*

*}*

*fclose($f);*

Если это не просмотр результатов, а голосование, производим необходимые действия для учета голоса.

Записывает IP адресс

*$ip\_adr = fopen("ip$id.dat","a++");*

*flock($ip\_adr,LOCK\_EX);*

*fputs($ip\_adr, "$ip".",");*

*fflush($ip\_adr);*

*flock($ip\_adr,LOCK\_UN);*

*fclose($ip\_adr);*

*}*

Печатает список ответов и результатов

*for ($i=1;$i<count($data);$i++) {*

*$votes = explode("~", $data[$i]);*

*echo "$votes[1]: <b>$votes[0]</b><br>";*

*}*

Код проверяет ip адреса в ip3.dat и записывает результат голосования в 3.dat фаил.

Обработка PHP осуществляется в OpenServere.

**Чтобы проголосовать нужно**:

* Зайти на сайт
* Открыть страничку “Опрос”
* И поставить любую оценку

Сначала PHP проверит ваш IP Адрес, если вы уже голосовали то сайт выдаст следующее сообщение “Вы уже голосовали!” определенным цветом.

Чтобы избежать проверки IP адреса и голосовать повторно нужно проделать следующее с кодом:

Просто закомментировать эти строки:

*foreach($ip\_abbr as $value)*

*if ($ip == $value) {echo "<p><b><font color=red> Вы уже голосовали! </font></b></p>";*

*exit;*

*}*

Повторное голосование разрешено.

Если просто закомментировать эти строки код не будет проверять ваш IP адрес, иными словами будет – Открытое голосование.

**Так же на сайте используется Javascript**

С помощью Javascript на сайте работает часы

В HTML документе прописывается js функция вызывающая скрипт часов

*<script language="javascript">*

*<!--*

*clock();*

*//-->*

*</script>*

Сама функция:

*function checklength(i){*

*if (i<10)*

*{i="0"+i;}*

*return i;*

*}*

*function clock(){*

*var now = new Date();*

*var hours = checklength(now.getHours());*

*var minutes = checklength(now.getMinutes());*

*var seconds = checklength(now.getSeconds());*

*var format = 1; //0=24 hour format, 1=12 hour format*

*var time;*

*if (format == 1){*

*if (hours >= 12){*

*if (hours ==12){*

*hours = 12;*

*}else {*

*hours = hours-12;*

*}*

*time=hours+':'+minutes+':'+seconds+' PM';*

*}else if(hours < 12){*

*if (hours ==0){*

*hours=12;*

*}*

*time=hours+':'+minutes+':'+seconds+' AM';*

*}*

*}*

*if (format == 0){*

*time= hours+':'+minutes+':'+seconds;*

*}*

*document.getElementById("clock").innerHTML=time;*

*setTimeout("clock();", 500);*

*}*

Описание функции

*function clock(){*

*var now = new Date();*

*var hours = checklength(now.getHours());*

*var minutes = checklength(now.getMinutes());*

*var seconds = checklength(now.getSeconds());*

*var format = 1; //0=24 hour format, 1=12 hour format*

*var time;*

Эта часть функции

* Создает функцию с названием clock()
* Создает 12 часовой формат
* Создает необходимые переменные

Эта часть функции

*if (format == 1){*

*if (hours >= 12){*

*if (hours ==12){*

*hours = 12;*

*}else {*

*hours = hours-12;*

*}*

*time=hours+':'+minutes+':'+seconds+' PM';*

*}else if(hours < 12){*

*if (hours ==0){*

*hours=12;*

*}*

*time=hours+':'+minutes+':'+seconds+' AM';*

*}*

*}*

* Проверяет, если переменные больше задаваемых значений, то отображаться будет PM
* Если же переменные меньше своих значений, то отображаться будет AM

Эта часть функции

*if (format == 0){*

*time= hours+':'+minutes+':'+seconds;*

*}*

*document.getElementById("clock").innerHTML=time;*

*setTimeout("clock();", 500);*

*}*

* Выполняет задержку в миллисекундах (1 секунда 1000 миллисекунд)
* Отображает часы на сайте

Так же на сайте расположено несколько блоков с определенными значениями

*<style>*

*html,body{*

*background-color:gray;/\*left, right Цвет фона веб-страницы \*/*

*background-image:url(./img/0.jpg );*

*<!--opacity:0.86;-->*

*position:fixed;*

*width:100%;*

*height:100%;*

*overflow-x: hidden;*

*overflow-y: hidden;*

*background-repeat: no-repeat;*

*background-position: center;*

*background-attachment:fixed;*

*}*

*</style>*

В каждой HTML странице прописаны свои универсальные стили, но есть и один общий стиль:

*.tel {*

*position:relative;*

*width:60%;*

*height:10%;*

*overflow-x: hidden;*

*overflow-y: hidden;*

*margin-left: 20%;*

*margin-top: 60%;*

*border-radius:2%;*

*border-style:solid;*

*background-color:white;*

*opacity:0.85;*

*text-align:center;*

*word-break:break-all;*

*}*

С этим блоком связанны все три страницы, но на каждой HTML странице прописаны свои параметры в связи с чем на всех трех страницах отображается что то своё.