

**Национальное образовательное частное учреждение
дополнительного профессионального образования «Национальный
открытый университет «ИНТУИТ»**

Допускаю к защите

Первый проректор

А.П. Сериков

« » 2020 г.

ВЫПУСКНАЯ РАБОТА

на тему «Разработка сайта ингушского перевода Библии»
(добавление динамических элементов к сайту)

Программа профессиональной переподготовки
«Профессиональное веб-программирование»

Исполнитель:

Мигунов А. Ю.

Руководитель:

Сериков А. П.

Москва

2020

Введение

Темой данной выпускной работы является разработка динамических элементов для ранее разработанного статического веб-сайта для публикации книг Библии на ингушском языке.

Данная работа посвящена разработке следующих новых элементов для веб-сайта: 1) раскрывающихся меню – с использованием JavaScript; 2) опции дополнительного просмотра глав на русском и чеченском языках – также реализованной с помощью JavaScript; 3) опции поиска по текстам, опубликованным на сайте (по выбору – на ингушском, русском или чеченском языках) – на основе JavaScript и PHP; 4) формы для обратной связи и скрипта для ее обработки – также с помощью JavaScript и PHP.

Глава 1. Разработка раскрывающихся меню

В рамках предыдущей ВКР были разработаны три навигационных меню – верхнее, левое (которое скрывается при просмотре сайта на смартфоне) и нижнее (которое скрыто при просмотре сайта на компьютере, но становится видимым при просмотре сайта на смартфоне). Верхнее меню обеспечивает переход только к главной странице сайта, а также к главным страницам (предисловиям) к книгам. Внутри каждой книги для перехода по главам используется левое или нижнее меню, а также ссылки на предыдущую и последующую главы в верхней части страницы.

Для более удобной навигации и ориентации по сайту меню можно сделать раскрывающимися, что и было сделано в данной части работы. Причем верхнее меню было решено сделать раскрывающимся с возможностью одновременного открытия всех подменю, а левое – в виде меню-аккордеона с возможностью раскрытия только одного подменю.

Для реализации раскрывающихся меню можно использовать чистый CSS без необходимости использования JavaScript. Причем это получается даже проще, чем с использованием JavaScript. Однако, поскольку целью данной ВКР является создание динамического сайта, было решено использовать JavaScript для создания раскрывающихся меню – в большей степени, в учебных целях, чем в практических.

1.1. Разработка верхнего раскрывающегося меню

Первый шаг – создание HTML кода нового меню. В отличие от старого варианта верхнего меню, основное меню выполнено в виде списка, а все подменю – в виде вложенных списков. В сокращенном виде HTML код имеет следующий вид:

```
<nav id="nav_top2">
  <ul class="topmenu">
    <li><a id="main_index" href="/index.html"
onMouseOver="ShowSubmenu('index')">Главная</a>
      <ul id="ind" class="submenu">
        <li><a href="/search.html">Поиск по сайту</a></li>
        <li><a href="/feedback.html">Обратная связь</a></li>
        <li><br /></li>
      </ul>
    </li>
    <li>|</li>
    <li><a id="gen_main" href="/genesis.html"
onMouseOver="ShowSubmenu('genesis')">Доладалар</a>
      <ul id="gen" class="submenu">
        <li><a href="/genesis1.html">Корта 1</a></li>
        <li><a href="/genesis2.html">Корта 2</a></li>
        .....
        <li><a href="/genesis50.html">Корта 50</a></li>
        <li><br /></li>
      </ul>
    </li>
    <li>|</li>
```

```

<li><a id="rut_main" href="./ruth.html"
onMouseOver="ShowSubmenu('ruth')">Рyфн</a>
    <ul id="rut" class="submenu">
        <li><a href="./ruth1.html">Копта 1</a></li>
        <li><a href="./ruth2.html">Копта 2</a></li>
        <li><a href="./ruth3.html">Копта 3</a></li>
        <li><a href="./ruth4.html">Копта 4</a></li>
        <li><br /></li>
    </ul>
</li>
<li>|</li>
<li><a id="est_main" href="./esther.html"
onMouseOver="ShowSubmenu('esther')">Эстаp</a>
    <ul id="est" class="submenu">
        <li><a href="./esther1.html">Копта 1</a></li>
        <li><a href="./esther2.html">Копта 2</a></li>
        .....
        <li><a href="./esther10.html">Копта 10</a></li>
        <li><br /></li>
    </ul>
</li>
<li>|</li>
<li><a id="pro_main" href="./proverbs.html"
onMouseOver="ShowSubmenu('proverbs')">Кицаш</a>
    <ul id="pro" class="submenu">
        <li><a href="./proverbs1.html">Копта 1</a></li>
        <li><a href="./proverbs2.html">Копта 2</a></li>
        .....
        <li><a href="./proverbs31.html">Копта 31</a></li>

```

```

<li><br /></li>

</ul>

</li>

<li>|</li>
<li><a id="jon_main" href="/jonah.html"
onMouseOver="ShowSubmenu('jonah')">Юнус-пайхамар</a>
  <ul id="jon" class="submenu">
    <li><a href="/jonah1.html">Корта 1</a></li>
    <li><a href="/jonah2.html">Корта 2</a></li>
    <li><a href="/jonah3.html">Корта 3</a></li>
    <li><a href="/jonah4.html">Корта 4</a></li>
    <li><br /></li>
  </ul>
</li>
</ul>
</nav>

```

Как видно из приведенного кода, к каждому основному пункту меню добавлен вызов пользовательской функции JavaScript под названием ShowSubmenu() по событию onMouseOver – то есть при наведении мыши на соответствующий пункт меню. Эта функция служит для раскрытия подменю, и при ее вызове ей передается параметр, который указывает, какое именно подменю должно быть раскрыто.

Второй шаг – создание скрипта на JavaScript для верхнего навигационного меню. Поскольку этот скрипт будет использоваться на всех страницах сайта, он вынесен в отдельный файл (в который также будет добавлен скрипт для левого навигационного меню) –

script_menus.js, а этот файл подключается ко всем страницам сайта с помощью кода внутри раздела head:

```
<script type="text/javascript" src="scripts_menus.js">  
</script>
```

Функция для раскрытия подменю достаточно простая. Она просто перебирает все возможные варианты и раскрывает соответствующее меню:

```
function ShowSubmenu(submenuName) {  
    if (submenuName == 'index') {  
        document.getElementById("ind").style.display = "block";  
    }  
    else if (submenuName == 'genesis') {  
        document.getElementById("gen").style.display = "block";  
    }  
    else if (submenuName == 'ruth') {  
        document.getElementById("rut").style.display = "block";  
    }  
    else if (submenuName == 'esther') {  
        document.getElementById("est").style.display = "block";  
    }  
    else if (submenuName == 'proverbs') {  
        document.getElementById("pro").style.display = "block";  
    }  
    else if (submenuName == 'jonah') {  
        document.getElementById("jon").style.display = "block";  
    }  
}
```

}

В этой функции нет варианта `else`. Это сделано из тех соображений, чтобы все имеющиеся возможности выбора подменю с главами книг передавались в функцию явно и не возникло проблем в дальнейшем, при добавлении новых книг.

Внутри каждого из вариантов выбора стиль CSS `display`, который был первоначально установлен как `none` меняется на `block`, и соответствующее подменю становится видимым.

Каждое подменю имеет свой идентификатор, который и служит для обращения к соответствующему подменю и изменения его стиля.

При загрузке страницы, а также при выходе за пределы верхнего меню все подменю должны быть закрыты. Для этого служит функция `HideAll()`:

```
function HideAll() {  
    document.getElementById("ind").style.display = "none";  
    document.getElementById("gen").style.display = "none";  
    document.getElementById("rut").style.display = "none";  
    document.getElementById("est").style.display = "none";  
    document.getElementById("pro").style.display = "none";  
    document.getElementById("jon").style.display = "none";  
}
```

Эта функция вызывается при загрузке страницы и при перемещении мыши в другую часть страницы.

Третий шаг – разработка кода CSS для верхнего меню:

nav#nav_top2 { // nav_top2 – это идентификатор для верхнего
раскрывающегося меню

margin-top: -15px;

margin-bottom: -10px;

margin-left: -40px;

text-align: center;

font-size: 16pt;

font-weight: bold;

}

ul.topmenu li {

display: inline-block; // все пункты основного меню размещаются в ряд,
как и раньше

position: relative;

}

ul.submenu {

position: absolute;

left: 0;

z-index: 5; // обеспечивается возможность наложения подменю друг на
друга

width: 100px;

background-color: #007600; // цвет фона для подменю выбран таким же,
как для верхнего меню – зеленым

text-align: left;

font-size: 12pt; // размер шрифта для подменю будет меньше, чем для
основного меню

font-weight: bold;

font-style: italic;

```
padding-left: 20px;  
padding-bottom: 20px;  
}
```

```
ul#ind {  
    width: 160px; // для первого подменю потребовалось установить  
    ширину побольше, поскольку пункты этого подменю длиннее  
}
```

```
ul.submenu li a {  
    color: yellow; // цвет текста пунктов подменю будет желтым  
}
```

1.2. Разработка левого меню-аккордеона

Код HTML почти не отличается от кода для верхнего меню:

```
<nav id="nav_side2">
  <ul class="topmenu2">
    <li><a id="main_index2" href="/index.html"
onMouseOver="ShowSubmenu2('index')"> Главная </a>
      <ul id="ind2" class="submenu2">
        <li><a href="/search.html">Поиск по сайту</a></li>
        <li><a href="/feedback.html">Обратная связь</a></li>
      </ul>
    </li>
    <li><a id="gen_main2" href="/genesis.html"
onMouseOver="ShowSubmenu2('genesis')"> Доладалап </a>
      <ul id="gen2" class="submenu2">
        <li><a href="/genesis1.html">Корта 1</a></li>
        .....
      </ul>
    </li>
    <li><a id="rut_main2" href="/ruth.html"
onMouseOver="ShowSubmenu2('ruth')"> Руфи </a>
      <ul id="rut2" class="submenu2">
        <li><a href="/ruth1.html">Корта 1</a></li>
        .....
      </ul>
    </li>
    <li><a id="est_main2" href="/esther.html"
onMouseOver="ShowSubmenu2('esther')"> Эстап </a>
      <ul id="est2" class="submenu2">
```

```

<li><a href="/esther1.html">Копта 1</a></li>
.....
</ul>
</li>
<li><a id="pro_main2" href="/proverbs.html"
onMouseOver="ShowSubmenu2('proverbs')"> Кицаш </a>
  <ul id="pro2" class="submenu2">
    <li><a href="/proverbs1.html">Копта 1</a></li>
    .....
  </ul>
</li>
<li><a id="jon_main2" href="/jonah.html"
onMouseOver="ShowSubmenu2('jonah')"> Юнус-пайхамар </a>
  <ul id="jon2" class="submenu2">
    <li><a href="/jonah1.html">Копта 1</a></li>
    .....
  </ul>
</li>
<li><br /></li>
</ul>
</nav>

```

Для раскрытия соответствующих подменю в левом меню используется функция ShowSubmenu2(), которой передается параметр, указывающий, какое подменю должно быть раскрыто. Эта функция немного отличается от функции ShowSubmenu(), служащей для раскрытия подменю верхнего меню, поскольку она должна не только раскрывать соответствующее подменю, но и закрывать все остальные:

```

function ShowSubmenu2(submenuName) {
    if (submenuName == 'index') {
        document.getElementById("ind2").style.display = "block";
    }
    else if (submenuName == 'genesis') {
        document.getElementById("gen2").style.display = "block";
    }
    else if (submenuName == 'ruth') {
        document.getElementById("rut2").style.display = "block";
    }
    else if (submenuName == 'esther') {
        document.getElementById("est2").style.display = "block";
    }
    else if (submenuName == 'proverbs') {
        document.getElementById("pro2").style.display = "block";
    }
    else if (submenuName == 'jonah') {
        document.getElementById("jon2").style.display = "block";
    }

    if (submenuName != 'index') {
        document.getElementById("ind2").style.display = "none";
    }
    if (submenuName != 'genesis') {
        document.getElementById("gen2").style.display = "none";
    }
    if (submenuName != 'ruth') {
        document.getElementById("rut2").style.display = "none";
    }
}

```

```

if (submenuName != 'esther') {
    document.getElementById("est2").style.display = "none";
}
if (submenuName != 'proverbs') {
    document.getElementById("pro2").style.display = "none";
}
if (submenuName != 'jonah') {
    document.getElementById("jon2").style.display = "none";
}
}

```

Первая половина этой функции практически аналогична функции ShowMenu(), а вторая ее половина закрывает все остальные подменю, кроме выбранного.

При загрузке страницы и перемещении мыши за пределы левого меню все подменю должны быть закрыты. Для этого используется функция HideAll2(), аналогичная функции HideAll() для верхнего меню. Эти две функции нельзя объединить вместе, поскольку они должны вызываться при разных перемещениях мыши.

```

function HideAll2() {
    document.getElementById("ind2").style.display = "none";
    document.getElementById("gen2").style.display = "none";
    document.getElementById("rut2").style.display = "none";
    document.getElementById("est2").style.display = "none";
    document.getElementById("pro2").style.display = "none";
    document.getElementById("jon2").style.display = "none";
}

```

Код CSS для левого меню-аккордеона:

```
nav#nav_side2 {  
    padding-left: 0;  
    margin-left: -40px;  
    margin-top: -20px;  
    text-align: left;  
    font-size: 14pt;  
    font-style: italic;  
    font-weight: bold;  
    position: absolute;  
    left: 30px;  
    top: 200px;  
}  
ul.submenu2 {  
    margin-left: -20px;  
}  
ul.submenu2 li a {  
    color: yellow;  
    font-size: 12pt;  
}
```

1.3. Дополнительные функции для обработки меню

При разработке статической версии сайта для облегчения ориентации посетителей изменялся цвет ссылок: желтый цвет – при нахождении на соответствующей странице, оранжевый цвет – при нахождении внутри соответствующей странице. Это было сделано с помощью различий в HTML коде для разных страниц. Однако можно сделать один и тот же HTML код для навигационных меню на всех страницах, а переключение цвета ссылок реализовать с помощью JavaScript, что и было сделано с помощью функции ThisPage():

```
function ThisPage() {  
    var thisName = document.location.pathname;  
    if (thisName.indexOf("index") > -1 ) { // ищется вхождение "index" в  
        имя файла. Если оно найдено, то значит, это главная страница, поэтому  
        цвет ссылок на главную страницу нужно изменить на желтый  
        document.getElementById("main_index").style.color = "yellow"; //  
        меняется цвет ссылки на верхнем меню  
        document.getElementById("main_index2").style.color = "yellow"; //  
        меняется цвет ссылки на левом меню  
    }  
    else if ((thisName.indexOf("feedback") > -1 ) ||  
        (thisName.indexOf("search") > -1 )) { // ищется вхождение "feedback" или  
        "search" в имя файла. feedback.html – это файл с формой для обратной  
        связи, а search.html – это файл с формой для поиска по текстам сайта.  
        Поскольку ссылки на эти страницы находятся в подменю главной  
        страницы, при нахождении пользователя на одной из этих страниц, цвет  
        ссылок на главную страницу должен стать оранжевым.  
        document.getElementById("main_index").style.color = "orange"; //  
        меняется цвет ссылки на верхнем меню
```



```

        document.getElementById("main_index2").style.color = "orange"; //
меняется цвет ссылки на левом меню
    }
    else if (thisName.indexOf("genesis") > -1 ) { // ищется вхождение
"genesis" – если оно найдено, это значит что посетитель находится на
одной из страниц книги Бытие (по-ингушски – Доладалар)
        if (thisName.indexOf("genesis.html") > -1) { // ищется вхождение
"genesis.html" – то есть файла с предисловием к этой книге – если оно
найдено, то посетитель находится на странице предисловия к этой книге,
и ссылки с названием книги в меню должны изменить цвет на желтый
            document.getElementById("gen_main").style.color = "yellow";
            document.getElementById("gen_main2").style.color = "yellow";
        } else { // если вхождение не найдено, то значит пользователь
находится на другой странице книги, и ссылки с названием книги в
меню должны стать оранжевыми. Для остальных книг осуществляются
аналогичные операции.
            document.getElementById("gen_main").style.color = "orange";
            document.getElementById("gen_main2").style.color = "orange";
        }
    }
    else if (thisName.indexOf("ruth") > -1 ) {
        if (thisName.indexOf("ruth.html") > -1) {
            document.getElementById("rut_main").style.color = "yellow";
            document.getElementById("rut_main2").style.color = "yellow";
        } else {
            document.getElementById("rut_main").style.color = "orange";
            document.getElementById("rut_main2").style.color = "orange";
        }
    }
}

```

```

else if (thisName.indexOf("esther") > -1 ) {
    if (thisName.indexOf("esther.html") > -1) {
        document.getElementById("est_main").style.color = "yellow";
        document.getElementById("est_main2").style.color = "yellow";
    } else {
        document.getElementById("est_main").style.color = "orange";
        document.getElementById("est_main2").style.color = "orange";
    }
}
else if (thisName.indexOf("proverbs") > -1 ) {
    if (thisName.indexOf("proverbs.html") > -1) {
        document.getElementById("pro_main").style.color = "yellow";
        document.getElementById("pro_main2").style.color = "yellow";
    } else {
        document.getElementById("pro_main").style.color = "orange";
        document.getElementById("pro_main2").style.color = "orange";
    }
}
else if (thisName.indexOf("jonah") > -1 ) {
    if (thisName.indexOf("jonah.html") > -1) {
        document.getElementById("jon_main").style.color = "yellow";
        document.getElementById("jon_main2").style.color = "yellow";
    } else {
        document.getElementById("jon_main").style.color = "orange";
        document.getElementById("jon_main2").style.color = "orange";
    }
}
}

```

Поскольку теперь изменение цвета ссылок осуществляется с помощью JavaScript, для пользователей, у которых отключен JavaScript, все ссылки будут белыми, и им будет сложнее ориентироваться на сайте. Кроме того, поскольку теперь все подменю раскрываются также с помощью JavaScript, им будет намного сложнее найти нужную главу.

Для того, чтобы сделать использование сайта более комфортным для пользователей с отключенным JavaScript, было решено сохранить для них старые навигационные меню (верхнее и левое). Таким образом, теперь каждая страница содержит по два верхних и по два левых меню. При загрузке страницы вначале с помощью CSS скрываются новые верхнее и боковое меню:

```
nav#nav_top2 {  
    display: none;  
}  
nav#nav_side2 {  
    display: none;  
}
```

После этого, если JavaScript включен, то при загрузке страницы вызывается функция, которая скрывает старые меню и делает видимыми новые меню, а также вызывает другие функции, которые требуется вызвать при загрузке страницы (HideAll(), HideAll2() и ThisPage()):

```
function ShowNavBars() {  
    document.getElementById("nav_top").style.display = "none";  
    document.getElementById("nav_top2").style.display = "block";  
    HideAll();
```

```
document.getElementById("nav_side").style.display = "none";  
document.getElementById("nav_side2").style.display = "block";  
HideAll2()  
ThisPage();  
}
```

Конечно, это не самый оптимальный вариант, поскольку он требует дополнительного HTML кода. Но с помощью него удалось, с одной стороны, использовать одинаковый HTML код в новых меню на всех страницах и сделать всю обработку новых меню с помощью JavaScript, а с другой стороны, сохранить сайт удобным для пользователей с отключенным JavaScript. Кроме того, для их удобства были добавлены дополнительные ссылки в левое меню – для перехода к форме обратной связи и форме поиска по сайту, поскольку в статической версии этого сайта этих страниц не было.

Раскрывающиеся меню можно было бы создать с помощью чистого CSS, не используя JavaScript, и это было бы проще в реализации. Но поскольку эта ВКР – по программе “профессиональное веб-программирование”, а не по программе “профессиональный веб-дизайн”, выбор JavaScript для создания раскрывающихся меню был обусловлен учебными целями.

Внешний вид сайта с раскрытыми навигационными меню приведен на рис.1 и 2.

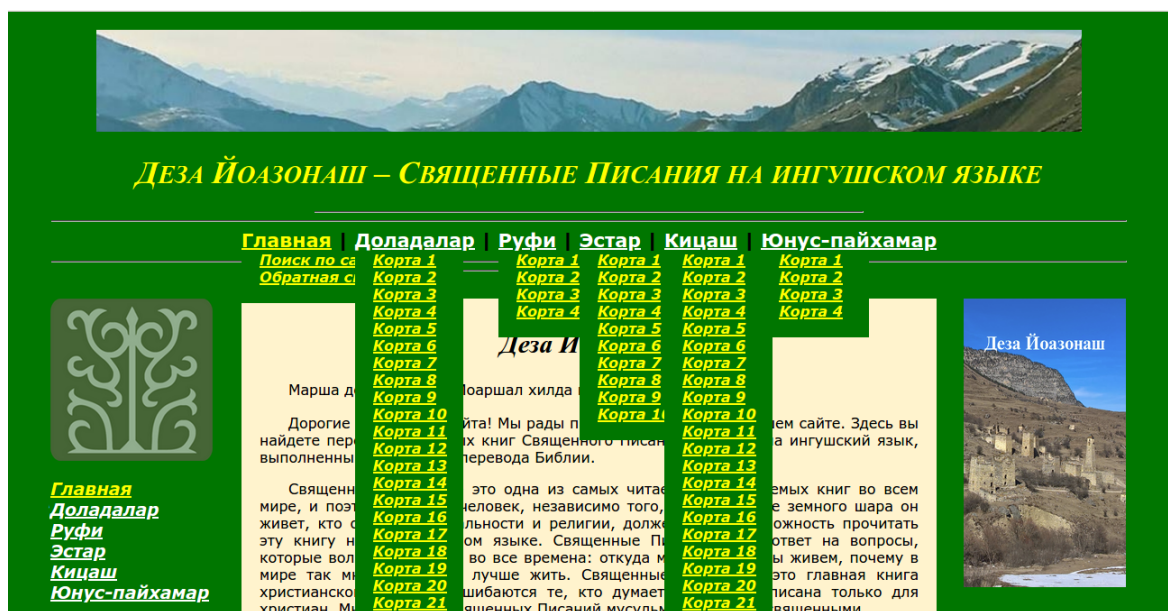


Рис. 1. Внешний вид главной страницы сайта при раскрытии всех подменю в верхнем навигационном меню

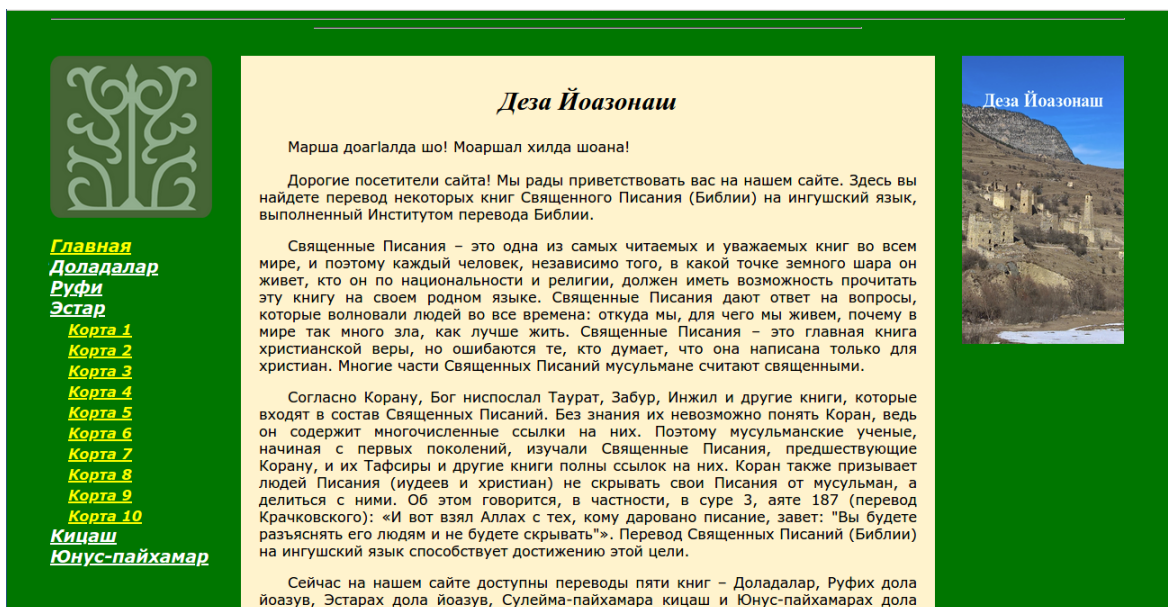


Рис. 2. Внешний вид главной страницы сайта при раскрытии одного из подменю в левом навигационном меню

Глава 2. Разработка системы подключения дополнительных переводов

В данном разделе была разработана опция просмотра русского и/или чеченского переводов одновременно с ингушским переводом. Для этого русский и чеченский переводы загружаются во внутренние фреймы на странице ингушского перевода той же самой главы.

Тексты русского и чеченского переводов были взяты из модулей для программы BibleQuote, которая позволяет читать различные переводы Библии на компьютере. Модули для этой программы включают в себя файлы библейских книг в формате HTML, но в упрощенном виде – в них полностью отсутствует раздел head, а также нет ряда других необходимых тегов.

Для использования на разрабатываемом веб-сайте эти файлы сначала были разделены на главы – по одной главе в каждом файле, а потом к ним были добавлены самые необходимые HTML теги:

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8" />
<title></title>
</head>
<body>
.....
</body>
</html>
```

Поскольку эти файлы будут, в основном, использоваться не самостоятельно, а подключаться как внутренние фреймы на страницах с

ингушским текстом, в более сложном форматировании HTML нет необходимости.

Все файлы получили имена, аналогичные соответствующим файлам на ингушском языке, но с добавлением “rus” для русского перевода и “che” для чеченского перевода.

Для добавления внутренних фреймов на каждую страницу, содержащую главу на ингушском языке, был добавлен следующий код HTML:

```
<p style="text-align:left; text-indent:0">
<!-- Добавляем кнопки для открытия и закрытия русского и чеченского
переводов. Нажатие на каждую из кнопок будет вызывать
соответствующую функцию. -->
<input type="button" value="Чеченский перевод"
onClick="ViewChechen();" />
<input type="button" value="Русский перевод"
onClick="ViewRussian();" />
<input type="button" value="Закрыть чеченский"
onClick="CloseChechen();" />
<input type="button" value="Закрыть русский"
onClick="CloseRussian();" />
</p>
<!-- Дополнительный код для пользователей, у которых отключен
JavaScript, и который будет скрыт для пользователей, у которых он
включен. -->
<noscript><p style="text-align:left; text-indent:0; color:blue; font-
style:italic">Ваш браузер не поддерживает JavaScript. Для просмотра
```

чеченского и русского переводов включите поддержку JavaScript или используйте ссылки.</p>

<!-- В данном случае это 1-я глава книги Бытие, поэтому страница чеченского перевода – genesis1che.html, а русского перевода - genesis1rus.html -->

Чеченский перевод

Русский перевод

</noscript>

<!-- Вставка внутренних фреймов, в которые будут загружаться чеченский и русский переводы. При загрузке страницы эти фреймы скрыты и становятся видимыми после нажатия на соответствующую кнопку. -->

<iframe id="che" style="display:none"></iframe>

<iframe id="rus" style="display:none"></iframe>

Скрипт на JavaScript для загрузки дополнительных переводов во внутренние фреймы для всех страниц, содержащих главы из книг на ингушском языке, будет один и тот же. Однако его не нужно подключать к тем страницам, где нет фреймов – главной странице, страницам с формами и страницам с предисловиям к книгам. Поэтому он помещен в отдельный файл – scripts.js и подключается к каждой странице, где есть фреймы с помощью кода в разделе head:

<script type="text/javascript" src="scripts.js">

</script>

Сам скрипт содержит четыре функции – по две для добавления переводов и по две для их отключения.

Функция ViewChechen() вызывается при нажатии на кнопку “Чеченский перевод” и выглядит так:

```
function ViewChechen() {  
    var fileLoc = document.location.href; // идентификация имени файла, с  
    которого вызывается функция  
    var fileNew = fileLoc.replace(".html", "che.html"); // поскольку имена  
    файлов с главами на чеченском языке отличаются от имен файлов с  
    соответствующими главами на ингушском языке только тем, что в конце  
    них добавлено “che”, добавляя эти буквы к названию текущего файла на  
    ингушском языке, получаем имя файла, который надо загрузить во  
    внутренний фрейм  
    document.getElementById("che").src = fileNew; // передаем ссылку на  
    нужный файл во внутренний фрейм  
    che.style.display = "inline"; // делаем фрейм с чеченским переводом  
    видимым, а если будут одновременно видимыми фреймы с чеченским и  
    русским переводами, то они будут расположены рядом друг с другом  
  
    if (rus.style.display == "none") { // проверяем, включен ли фрейм с  
    русским переводом  
        che.style.width = "90%"; // если фрейм с русским переводом отключен,  
        то растягиваем фрейм с чеченским переводом почти на всю ширину  
        основной части страницы (центральной колонки)  
    }  
    else if (rus.style.display == "inline") { // если фрейм с русским переводом  
    включен, то для обоих фреймов (с русским и чеченским переводом)
```

отводим равную ширину, которая будет немного меньше, чем половина ширины основной части страницы

```
rus.style.width = "45%";  
che.style.width = "45%";  
}
```

if (document.documentElement.clientHeight) { // проверяем, поддерживает ли браузер свойство documentElement.clientHeight, поскольку некоторые старые браузеры его не поддерживают

if (document.documentElement.clientHeight < 600) { // если высота видимой части экрана менее 600 px (что в первую очередь касается смартфонов), то устанавливаем высоту фрейма в половину этой высоты, чтобы помимо дополнительных переводов был видим основной текст на ингушском языке

```
var height = 0.5 * document.documentElement.clientHeight;  
che.style.height = height + "px";  
}
```

else { // если высота видимой части экрана равна или больше 600 px, то устанавливаем высоту фрейма 300 px, чтобы он не занимал слишком много места на экране

```
che.style.height = "300px";  
}  
}
```

else { // если браузер не поддерживает свойство documentElement.clientHeight, то просто устанавливаем высоту фрейма 300 px

```
che.style.height = "300px";  
}  
}
```

Функция ViewRussian(), которая показывает русский перевод при нажатии на кнопку “Русский перевод”, в основном, работает аналогично функции ViewChechen():

```
function ViewRussian() {  
    var fileLoc = document.location.href;  
    var fileNew = fileLoc.replace(".html", "rus.html"); // для получения имени  
    файла на русском языке к имени файла на ингушском языке  
    добавляются буквы “rus”  
    document.getElementById("rus").src = fileNew;  
    rus.style.display = "inline";  
  
    if (che.style.display == "none") {  
        rus.style.width = "90%";  
    }  
    else if (che.style.display == "inline") {  
        che.style.width = "45%";  
        rus.style.width = "45%";  
    }  
  
    if (document.documentElement.clientHeight) {  
        if (document.documentElement.clientHeight < 600) {  
            var height = 0.5 * document.documentElement.clientHeight;  
            rus.style.height = height + "px";  
        }  
        else {  
            rus.style.height = "300px";  
        }  
    }  
}
```

```

    }
    else {
        rus.style.height = "300px";
    }
}

```

Функция CloseChechen() закрывает фрейм с чеченским переводом и вызывается при нажатии на клавишу “Заккрыть чеченский”:

```

function CloseChechen() {
    che.style.display = "none"; // фрейм с чеченским текстом делается
НЕВИДИМЫМ
    if (rus.style.display == "inline") { // если фрейм с русским переводом
открыт, то он растягивается до ширины 90%
        rus.style.width = "90%";
    }
}

```

Функция CloseRussian() закрывает фрейм с русским переводом и вызывается при нажатии на клавишу “Заккрыть русский”. Она работает аналогично функции CloseChechen():

```

function CloseRussian() {
    rus.style.display = "none";
    if (che.style.display == "inline") {
        che.style.width = "90%";
    }
}

```

Внешний вид страницы сайта при открытии одного или двух дополнительных переводов показан на рис. 3 и 4.

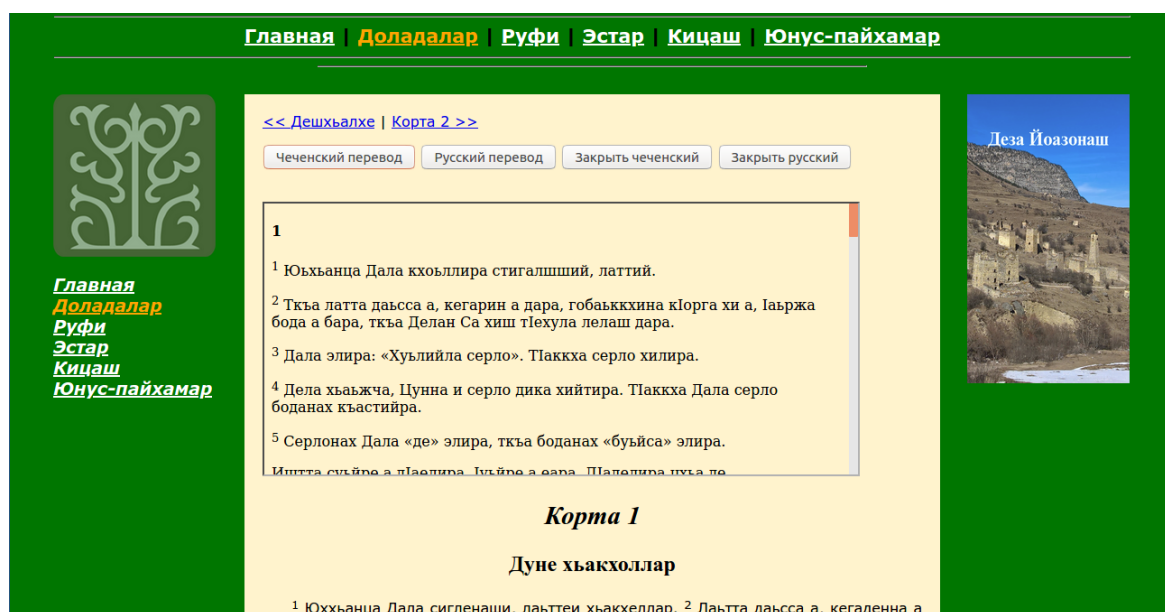


Рис. 3. Страница сайта с 1-й главой книги Бытие и открытым чеченским переводом

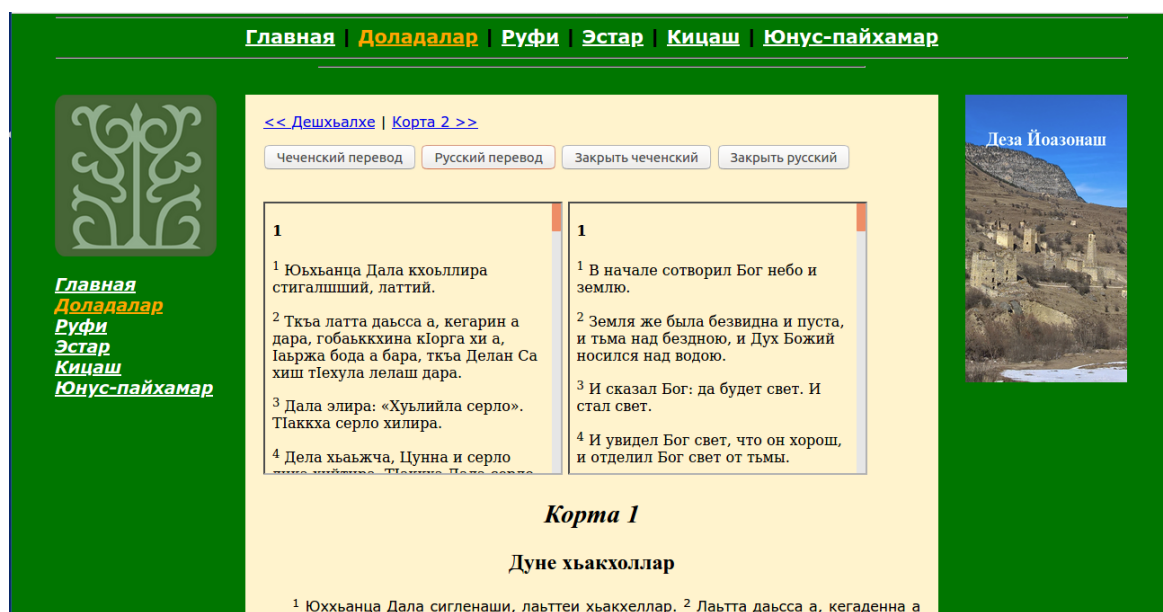


Рис. 4. Страница с 1-й главой книги Бытие с открытыми чеченским и русским переводами

Глава 3. Разработка системы поиска по текстам на сайте

В этом разделе была разработана система, дающая возможность для посетителей сайта найти стихи из библейских переводов (на одном из языков по выбору), в которых встречается то или иное слово. Она состоит из следующих компонентов: 1) форма для введения поискового запроса; 2) скрипт на JavaScript, проверяющий правильность введения данных в форму; 3) скрипт на PHP, который также проверяет правильность введения данных в форму (поскольку у пользователей может быть отключен JavaScript), а после этого обрабатывает поисковый запрос и выдает результаты; 4) файлы, по которым ведется поиск.

Для ускорения поиска по текстам сайта вначале были подготовлены специальные текстовые файлы, по которым и будет осуществляться поиск. Таким образом, вместо поиска по 99 имеющимся в настоящий момент отдельным главам (для каждого из трех языков) он будет осуществляться по одному файлу, в который помещены все главы, что значительно ускоряет поиск.

При обработке поискового запроса скрипт выдает список стихов, в которых встречается соответствующее слово, в котором каждый из пунктов списка отформатирован следующим образом:

```
<p><a href="ссылка на файл, содержащий соответствующую главу"
target="_blank"><b>Название книги, номер главы и номер стиха</b></a>
Текст стиха</p>
```

Поэтому для ускорения поиска можно сразу же соответствующим образом отформатировать все стихи в текстовых файлах для поиска. Тогда скрипт на PHP для обработки поискового запроса получается очень простым и соответственно работающим очень быстро.

Для создания текстовых файлов для поиска со специальным форматированием были созданы скрипты на Python.

В качестве примера приведу фрагмент скрипта на Python, использованный для создания текстового файла с русским текстом:

```
import re # подключаем модуль регулярных выражений

f1 = open("russian.txt", "w") # создаем файл для записи
f2 = open("01_genesis.htm", "r") # открываем файл с русским текстом
книги Бытие

chapter = 0
verse = 0

for line in f2: # обрабатываем каждую строку по отдельности
    text = line
    text = re.sub("<h4>|<p>|<sup>", "", text) # удаляем ненужные теги
    HTML в начале строки

    number1 = text.find("</h4>") # если строка содержит маркер h4, то это
    означает, что она содержит номер главы
    if (number1 != -1): # если это так, то увеличиваем номер главы на 1, а
    номеру стиха присваиваем значение 0, поскольку это начало новой
    главы
        chapter = chapter + 1
        verse = 0

    number2 = text.find("</sup>") # если строка содержит маркер sup, то это
    означает, что она содержит номер стиха
```

```

if (number2 != -1): # если это так, то увеличиваем номер стиха на 1
    verse = verse + 1

text = re.sub("</sup>|</p>|<p/>|<br/>", "", text) # убираем лишние теги
HTML из текста

number3 = text.find("h4") # снова проверяем, является ли данная строка
номером главы

if (number3 == -1): # если это не так, то данная строка будет записана в
текстовый файл, а если это номер главы, то она будет пропущена

    text1 = text # создаем новую переменную, в которую копируем
содержание соответствующего стиха

    link = "genesis" + str(chapter) + ".rus.html" # формируем имя файла, в
котором содержится соответствующая глава

    reference = "Доладалар (Бытие) " + str(chapter) + ":" + str(verse) #
формируем библийскую ссылку содержащую название книги, номер
главы и номер стиха

    text = "<p><a href=\"\" + link + \"\" target=\"_blank\"><b>" + reference +
"</b></a>\" \" + text1 # собираем всё вместе – ссылку на
соответствующую главу, библийскую ссылку и текст стиха

    f1.write(text) # записываем всё это в текстовый файл в виде одной
строки

f2.close()
f1.close()

```

Для записи остальных книг был создан аналогичный код на Python, который был добавлен к этому фрагменту. Эти фрагменты кода для других книг отличались от приведенного тем, что текстовый файл, в

который производилась запись открывался не для записи (с параметром “w”), а для добавления (с параметром “a”), для чтения открывались файлы с текстами соответствующих книг, а также тем, что имена файлов с текстами глав и библейские ссылки содержали соответствующее название книги.

Скрипт на Python для создания текстового файла для поиска с чеченским текстом был практически аналогичным, отличаясь только именем текстового файла для записи и именами файлов глав.

Скрипт на Python для создания текстового файла с ингушским текстом несколько отличался, что было связано с тем, что для его создания использовались файлы, полученные с помощью программы USFM Quick Converter, которые предварительно были модифицированы. Их HTML код отличался от кода файлов с русскими и чеченскими текстами.

Фрагмент этого скрипта имел следующий вид:

```
import re
```

```
f1 = open("ingush.txt", "w")
```

```
f2 = open("genesis.html", "r")
```

```
chapter = 0
```

```
for line in f2:
```

```
    text = line
```

```
    reg_exp1 = re.compile(r"^<span class=chaptermarker>[0-9-]+</span>$") #
```

```
    создаем регулярное выражение для поиска номера главы
```

if (reg_exp1.search(text)): # если найдено соответствие регулярному выражению, то эта строка содержит номер главы, поэтому к номеру главы добавляем 1

chapter = chapter + 1

reg_exp2 = re.compile(r"^[0-9-]+") # аналогично создаем регулярное выражение для поиска номера стиха

if (reg_exp2.search(text)): # поскольку в ингушском тексте некоторые номера стихов объединены вместе, нельзя просто прибавить 1 к номеру стиха, как в предыдущем скрипте; вместо этого надо “вытащить” этот номер из строки

text = text.replace("", "") # в случае, если строка содержит номер стиха, сначала убираем тег в начале строки

list_line = text.split("<") # далее разделяем строку на две части, используя первый символ закрывающего тега в качестве разделителя

verse = list_line[0] # помещаем первую часть строки (которая является номером стиха) в специальную переменную

text = list_line[1] # помещаем вторую часть строки (то есть, текст стиха) в другую переменную

text = text.replace("/span>", "") # удаляем оставшуюся часть закрывающего тега из начала строки, в которой содержится текст стиха

number1 = text.find("chaptermarker") # снова проверяем, содержит ли строка номер главы (поскольку этот тег не был удален)

if (number1 == -1): # если строка не содержит номер главы, то ее нужно будет записать в текстовый файл; оставшаяся часть этого скрипта работает аналогично скрипту для формирования текстового файла с русским текстом

text1 = text

```

link = "genesis" + str(chapter) + ".html"
reference = "Доладалар (Бытие) " + str(chapter) + ":" + str(verse)

text = "<p><a href=\"\" + link + \"\" target=\"_blank\"><b>" + reference +
"</b></a>\"" + text1

f1.write(text)
f1.close()
f2.close()

```

Таким образом, были созданы специальные текстовые файлы для поиска по текстам сайта.

После этого была создана простая форма для поиска:

```

<form name="search" action="/find.php" method="post" style="font-style:
italic">
<h3>Форма для поиска по сайту</h3>
<p>Выберите язык текста из списка:</p>
<p><select name="language" size="1">
<option value="none" selected>Выберите язык
<option value="ingush">Ингушский (гIалгIай мотт)
<option value="chechen">Чеченский (нохчийн мотт)
<option value="russian">Русский
</select></p>
<p>Введите слово или предложение для поиска (не более 200
символов):</p>
<p><input type="text" name="word" size="40" maxlength="200" value="">
</p>

```

```
<p><input type="submit" name="submit" value="Отправить"
onClick="return validate(this.form)" />&nbsp;&nbsp;&nbsp;<input type="reset"
name="reset" value="Очистить" /></p>
</form>
```

Далее была написана простая функция для валидации введенных данных – проверки, что значения введены:

```
function validate (form) {
  if (form.language.value == "none") {
    alert('Пожалуйста, выберите язык текста');
    return false;
  }

  if (!form.word.value) {
    alert('Пожалуйста, введите слово для поиска');
    return false;
  }

  form.word.value = htmlspecialchars(form.message_text.word);
  return true;
}
```

Эта функция вызывает другую функцию – htmlspecialchars(), которая преобразует специальные символы в HTML сущности. Это делается из соображений безопасности, чтобы пользователи не могли передать HTML код, а тем более скрипт на JavaScript, через форму поиска на сервер.

В отличие от PHP, в JavaScript такой библиотечной функции нет. Поэтому была создана пользовательская функция для выполнения этой задачи:

```
function htmlspecialchars(html) {  
    html = html.replace(/&/g, "&amp;");  
    html = html.replace(/</g, "&lt;");  
    html = html.replace(/>/g, "&gt;");  
    html = html.replace(/"/g, "&quot;");  
    html = html.replace(/'/g, "&apos;");  
    return html;  
}
```

Поскольку валидация формы обратной связи с помощью JavaScript требуется только для файла, в котором содержится эта форма, этот скрипт был добавлен в тот же файл, в раздел head.

Далее был разработан скрипт на PHP для обработки формы поиска. Поскольку поиск осуществляется по заранее подготовленным и отформатированным текстовым файлам, то скрипт получился относительно небольшой:

```
<?php  
// Прежде всего, осуществляется проверка, были ли введены данные в  
форму, а также преобразование специальных символов в HTML  
сущности. Такая повторная проверка на сервере требуется, поскольку у  
пользователей может быть отключен JavaScript и тогда форма для  
поиска будет послана на сервер без предварительной проверки.  
$err = "";  
if ($_POST["language"] == "none") {
```

```
$err .= "Не выбран язык для поиска." . "<br />";  
}
```

```
if (!$_POST["word"]) {  
    $err .= "Не введено слово для поиска." . "<br />";  
}
```

```
if ($err) {  
    echo $err;  
    echo "Пожалуйста, вернитесь назад и внесите изменения.";  
}
```

```
$word_search = $_POST["word"];  
$word_search = htmlspecialchars($word_search, ENT_QUOTES);
```

// Далее осуществляется непосредственная обработка формы. В кавказских языках (в том числе, в ингушском и чеченском) есть дополнительная буква, так называемая кириллическая палочка, которая визуально выглядит, как римская цифра “один” или заглавная латинская “i”. С ней возникают определенные проблемы, поскольку есть несколько похожих символов, которые могут быть использованы пользователями при вводе слов на ингушском или чеченском языке. Поэтому предварительно надо преобразовать эти символы в символ кириллической палочки.

```
$word_search = preg_replace("(I|I1|I|l)", "I", $word_search);
```

// Далее открывается для чтения текстовый файл с текстом на выбранном языке, который будет использоваться для поиска введенного пользователем слова.

```

$fp = "";
if ($_POST["language"] == "ingush") {
    $fp = @fopen("ingush.txt", "r") or die("Возникла ошибка, запрос не
может быть обработан");
}
else if ($_POST["language"] == "chechen") {
    $fp = @fopen("chechen.txt", "r") or die("Возникла ошибка, запрос не
может быть обработан");
}
else if ($_POST["language"] == "russian") {
    $fp = @fopen("russian.txt", "r") or die("Возникла ошибка, запрос не
может быть обработан");
}

```

// Если файл был успешно открыт, то скрипт осуществляет чтение его строка за строкой и ищет вхождение введенного пользователем слова в каждой строке. Если слово найдено, то данная строка выводится на экран.

```

if ($fp) {
    while(!feof($fp)) {
        $line = fgets($fp);
        if (strstr($line, $word_search)) {
            echo $line . "</p>"; // При выводе на экран добавляется закрывающий
тег абзаца. Это связано с тем, что текстовые файлы не содержат этих
тегов: скрипты на Python помещали эти теги на следующую строку
(поскольку они следовали за символом конца строки - \n), а это вызвало
бы проблемы при поиске в этих файлах, когда текст читается строка за
строкой, поэтому закрывающие теги были убраны из скриптов на Python
и добавлены в данный скрипт для поиска на РНР.

```

```

    }
}
fclose($fp);
}
?>

```

Работа системы поиска по сайту была протестирована на сервере ХАМРР (точнее, на LАМРР, то есть версии для Linux). На рис. 5 и 6 показан пример ее работы.

Рис. 5. Форма для поиска с введенными данными

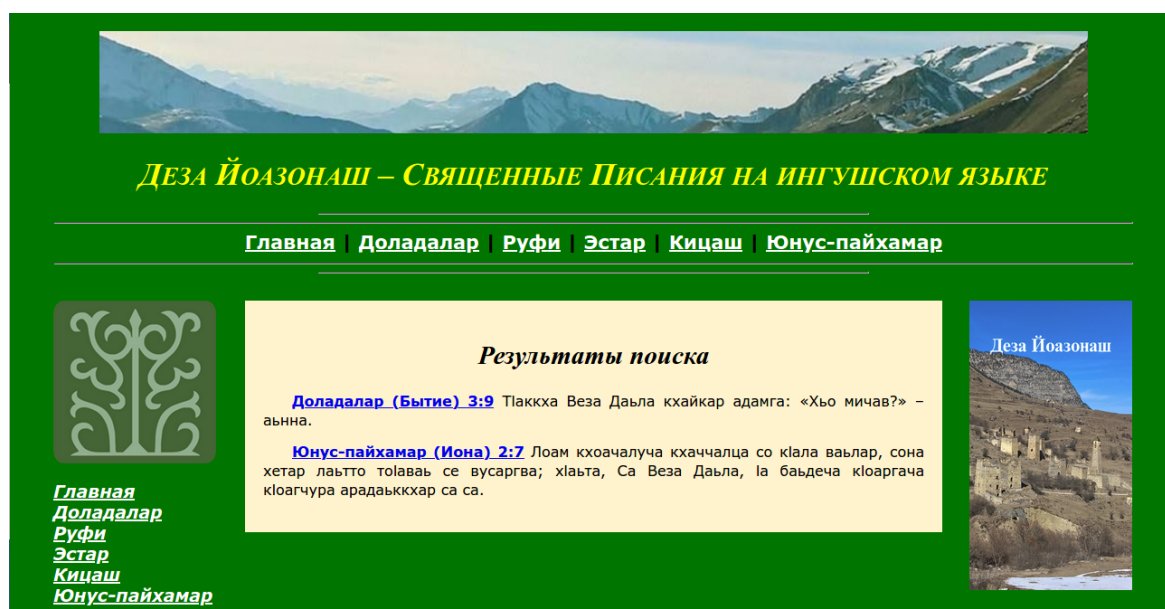


Рис. 6. Результаты поиска, полученные при данных, которые были введены в форму для поиска согласно рис. 5.

Глава 4. Разработка системы обратной связи с посетителями сайта

В этом разделе была создана форма для обратной связи и скрипты для ее обработки.

HTML код формы для обратной связи выглядит так:

```
<form name="feedback" action="./send.php" method="post" style="font-  
style: italic">  
<h3>Форма для обратной связи</h3>  
<p>(* Поля, обязательные для заполнения)</p><br />  
<p>* Тема сообщения:</p>  
<p>  
<select name="subject" size="1">  
<option value="none" selected>Выберите тему  
<option value="Вопрос">Вопрос  
<option value="Комментарий/замечание">Комментарий/замечание  
<option value="Предложение/пожелание">Предложение/пожелание  
<option value="Сообщение об ошибке на сайте">Сообщение об ошибке  
на сайте  
<option value="Другое">Другое  
</select></p>  
<p>* Имя:  
<input type="text" name="first_name" size="20" maxlength="60" value="">  
</p>  
<p>Отчество:  
<input type="text" name="middle_name" size="20" maxlength="60"  
value="">  
</p>
```

```

<p>Фамилия:
<input type="text" name="last_name" size="20" maxlength="60" value="">
</p>
<p>* Адрес электронной почты:</p>
<p>
<input type="text" name="email_address" size="20" maxlength="60"
value="">
</p>
<p>* Текст сообщения (не более 2000 символов):</p>
<p><textarea cols="50" rows="5" name="message_text" wrap="virtual"
maxlength="2000"></textarea>
</p>
<p>Как вы оцениваете сайт по пятибальной шкале:</p>
<p>
<input type="radio" name="eval" value="5">5
<input type="radio" name="eval" value="4">4
<input type="radio" name="eval" value="3">3
<input type="radio" name="eval" value="2">2
<input type="radio" name="eval" value="1">1
<input type="radio" name="eval" value="Без оценки" checked>Без оценки
</p>
<p><input type="submit" name="submit" value="Отправить"
onClick="return validate(this.form)" />&nbsp;&nbsp;&nbsp;<input type="reset"
name="reset" value="Очистить" /></p>
</form>

```

К этому же файлу был добавлен скрипт на JavaScript для валидации формы и преобразования специальных символов в HTML сущности.

```
<script language="JavaScript" type="text/javascript">
```

```
function htmlspecialchars(html) {  
    html = html.replace(/&/g, "&amp;");  
    html = html.replace(/</g, "&lt;");  
    html = html.replace(/>/g, "&gt;");  
    html = html.replace(/"/g, "&quot;");  
    html = html.replace(/'/g, "&apos;");  
    return html;  
}
```

```
function validate (form) {  
    if (form.subject.value == "none") { // Проверка, выбрана ли тема  
сообщения  
        alert('Пожалуйста, выберите тему сообщения');  
        return false;  
    }  
}
```

```
if (!form.first_name.value) { // Проверка, введено ли имя  
    alert('Пожалуйста, введите имя');  
    return false;  
}
```

// Проверка правильности введения имени. Надо заметить, что пользователь может ввести свое имя на ингушском языке и использовать разные символы для введения кириллической палочки, в том числе и цифру “1”, поэтому надо предусмотреть все такие варианты. Кроме того, он может ввести свое имя не кириллицей, а латинскими буквами.

```
var str1 = form.first_name.value;
```

```
var re1 = /^[a-zA-Za-яА-ЯёЁИИІІ\-.]+$/;
```

```
if(!re1.test(str1)) {
```

```
    alert('Пожалуйста, проверьте правильность введения имени');
```

```
    return false;
```

```
}
```

// Проверка правильности введения отчества. Поскольку в данной форме отчество является опциональным, то его наличие не проверяется.

```
var str2 = form.middle_name.value;
```

```
var re2 = /^[a-zA-Za-яА-ЯёЁИИІІ\-.]+$/;
```

```
if(!re2.test(str2) && str2) {
```

```
    alert('Пожалуйста, проверьте правильность введения отчества');
```

```
    return false;
```

```
}
```

// Проверка правильности введения фамилии, которая также является факультативной.

```
var str3 = form.last_name.value;
```

```
var re3 = /^[a-zA-Za-яА-ЯёЁИИІІ\-.]+$/;
```

```
if(!re3.test(str3) && str3) {
```

```
    alert('Пожалуйста, проверьте правильность введения фамилии');
```

```
    return false;
```

```
}
```

// Проверка, введен ли email.

```
if (!form.email_address.value) {
```

```
    alert('Пожалуйста, введите адрес email');
```

```
    return false;
```

```
}
```

// Проверка правильности введения email с использованием регулярного выражения

```
var str4 = form.email_address.value;  
var re4 = /^[^@]+@([a-z0-9\-\+\.])+[a-z]{2,4}$/;  
if(!re4.test(str4)) {  
    alert('Пожалуйста, проверьте правильность введения адреса email');  
    return false;  
}
```

// Проверка, введено ли сообщение

```
if (!form.message_text.value) {  
    alert('Пожалуйста, введите текст сообщения');  
    return false;  
}
```

// Преобразование специальных символов в HTML сущности

```
form.message_text.value = htmlspecialchars(form.message_text.value);  
return true;  
}  
</script>
```

Далее был разработан скрипт на PHP для обработки формы. Этот скрипт вначале повторно проверяет введенные данные (на случай, если у пользователя был отключен JavaScript и первичная проверка не была произведена), а после этого формирует из введенных данных сообщение и отправляет его администратору сайта по адресу, указанному в скрипте, а затем выдает пользователю сообщение о том, успешно ли была

отправлена его форма, и напоминает те данные, которые он ввел в форму.

```
<?php // Проверка данных, введенных в форму
$err = "";

if ($_POST["subject"] == "none") {
    $err .= "Вы не выбрали тему сообщения." . "<br />";
}

if (!$_POST["first_name"]) {
    $err .= "Вы не ввели имя." . "<br />";
}

else {
    $str1 = $_POST["first_name"];
    $re1 = "/[a-zA-Za-яA-ЯёЁІІіІ\~]+/";
    if(!(preg_match($re1, $str1))) {
        $err .= "Вы неправильно ввели имя." . "<br />";
    }
}

$str2 = "";
$str2 = $_POST["middle_name"];
if ($str2) {
    $re2 = "/[a-zA-Za-яA-ЯёЁІІіІ\~]+/";
    if(!(preg_match($re2, $str2))) {
        $err .= "Вы неправильно ввели отчество." . "<br />";
    }
}
```

```

}

$str3 = "";
$str3 = $_POST["last_name"];
if ($str3) {
    $re3 = "/[a-zA-Za-яА-ЯёЁИИІІ\~]+/";
    if(!preg_match($re3, $str3)) {
        $err .= "Вы неправильно ввели фамилию." . "<br />";
    }
}

if (!$_POST["email_address"]) {
    $err .= "Вы не ввели адрес email." . "<br />";
}

else {
    $str4 = $_POST["email_address"];
    $re4 = "/^[^@]+@([a-z0-9\~]+\.)+[a-z]{2,4}$/";
    if(!preg_match($re4, $str4)) {
        $err .= "Вы неправильно ввели адрес email." . "<br />";
    }
}

if (!$_POST["message_text"]) {
    $err .= "Вы не ввели текст сообщения." . "<br />";
}

if ($err) {
    echo $err;
}

```



```

echo "Пожалуйста, вернитесь назад и внесите исправления.";
die();
}

$_POST["message_text"] = htmlspecialchars($_POST["message_text"],
ENT_QUOTES);

// Если данные введены корректно, начинается обработка формы и
формируется сообщение, которое будет отправлено администратору
сайта по электронной почте.
$FirstName = $_POST["first_name"];
$MiddleName = $_POST["middle_name"];
$LastName = $_POST["last_name"];
>Email = $_POST["email_address"];
$Subject = $_POST["subject"];
$Message = $_POST["message_text"];
$Evaluation = $_POST["eval"];

$Body = "Пользователь: ";

$FullName = $FirstName;
if ($MiddleName) {
    $FullName = $FullName . " " . $MiddleName ;
}
if ($LastName) {
    $FullName = $FullName . " " . $LastName;
}

```

```
// Формирование сообщения, которое будет послано по электронной
почте
```

```
$Body = $Body . $FullName;
```

```
$Body = $Body . "\n" . "Email: " . $Email . " ";
```

```
$Body = $Body . "\n" . "Тема: " . $Subject . " ";
```

```
$Body = $Body . "\n" . "Текст сообщения: " . $Message . " ";
```

```
$Body = $Body . "\n" . "Оценка сайта: " . $Evaluation . " ";
```

```
// Отправка сообщения. Здесь нужно указать работающий email для
получения писем с сайта.
```

```
$formsent = mail("xampp@text.com", "Новое сообщение с ингушского
библейского сайта", $Body);
```

```
// Проверка успешности отправки и выдача соответствующего
сообщения пользователю
```

```
if ($formsent) {
```

```
    echo "Здравствуйте, $FullName! Мы получили Ваше сообщение!
```

```
    Спасибо, что Вы написали нам!";
```

```
}
```

```
else {
```

```
    echo "К сожалению, возникла проблема с отправкой Вашей формы.
```

```
    Попробуйте отправить ее снова.";
```

```
}
```

```
// Вывод сообщения пользователю о тех данных, которые он ввел.
```

```
$Body_echo = str_replace("\n", "<br />", $Body);
```

```
echo "<br />Вы ввели: " . $Body_echo;
```

```
?>
```

Для тестирования работы этой системы был снова использован сервер LAMPP. Поскольку в его состав не входит почтовый сервер, его пришлось настроить специальным образом, установив “почтовую заглушку”, чтобы можно было протестировать посылку сообщений с сайта, а также настроить почтовую программу (Evolution) для получения пусем с сервера.

Тестирование прошло успешно. Сообщение было доставлено по указанному в скрипте адресу. На рис. 7, 8 и 9 приведены результаты тестирования.

Дорогие посетители сайта! Вы можете воспользоваться формой для обратной связи, чтобы написать нам свои отзывы, комментарии, предложения, пожелания, задать вопросы.

Форма для обратной связи

(* Поля, обязательные для заполнения)

* Тема сообщения:

* Имя:

Отчество:

Фамилия:

* Адрес электронной почты:

* Текст сообщения (не более 2000 символов):

Как вы оцениваете сайт по пятибальной шкале:
☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 ☒ Без оценки

Рис. 7. Форма для обратной связи с введенными данными

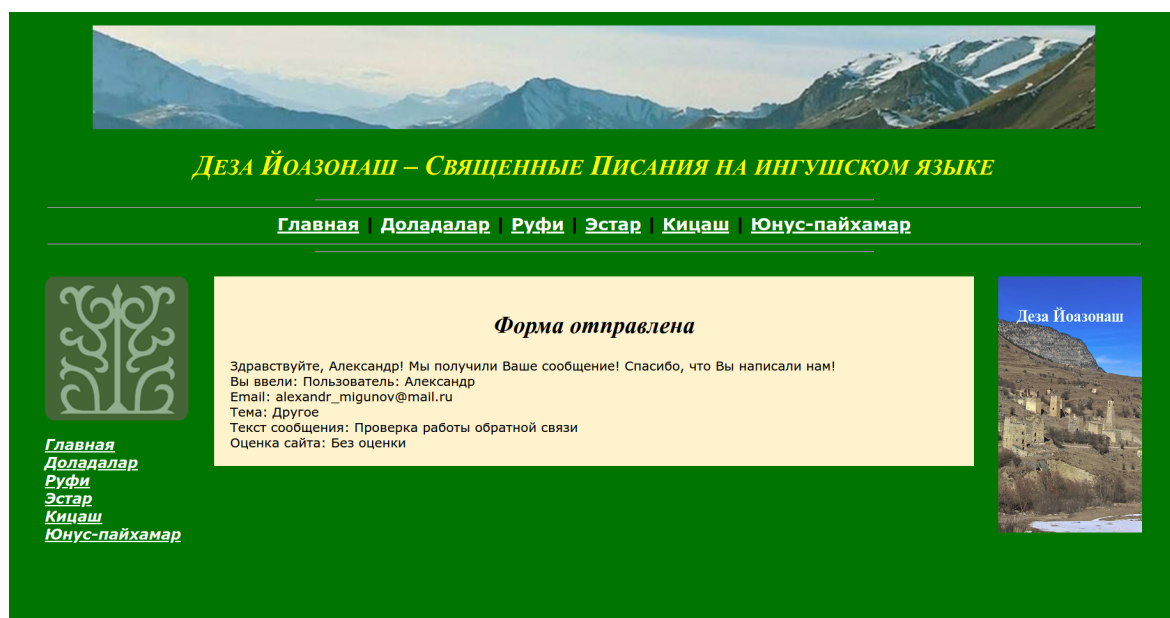


Рис. 8. Страница с сообщением, полученным после отправки формы

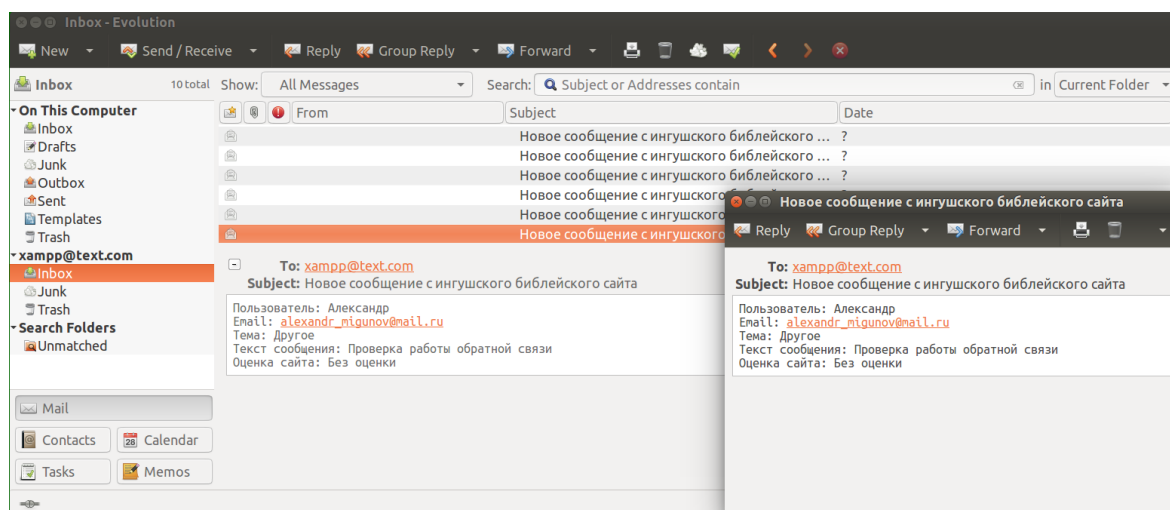


Рис. 9. Сообщение, полученное по электронной почте

Заключение

В результате проведенной работы к разработанному ранее статическому сайту для публикации библейских переводов на ингушском языке, были добавлены четыре новых динамических элемента: 1) раскрывающиеся меню – с использованием JavaScript; 2) опция дополнительного просмотра глав на русском и чеченском языках – также реализованная с помощью JavaScript; 3) опция поиска по текстам, опубликованным на сайте (по выбору – на ингушском, русском или чеченском языках) – на основе JavaScript и PHP; 4) форма для обратной связи и скрипт для ее обработки – также с помощью JavaScript и PHP.

Работа всех новых элементов сайта была проверена с помощью сервера LAMPP.

Список использованной литературы

1. Гудман Д. JavaScript. Библия пользователя, 5-е изд. / Д. Гудман, М. Моррисон. – Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2006. – 1184 с.: ил.
2. Пауэлл Т. Полный справочник по JavaScript, 2-е изд. / Т. Пауэлл, Ф. Шнайдер. Пер. с англ. – М.: Издательский дом “Вильямс”, 2007. – 960 с.: ил.
3. Васильев А.Н. JavaScript в примерах и задачах. / А.Н. Васильев. – М.: Эксмо, 2018. – 720 с.
4. Конверс Т. PHP5 и MySQL. Библия пользователя. / Т. Конверс, Дж. Парк, К. Морган. – Пер. с англ. – М.: Издательский дом “Вильямс”, 2007. – 1216 с.: ил.
5. Кузнецов М.В. Самоучитель PHP7 / М.В. Кузнецов, И.В. Симдянов. – СПб.: БХВ-Петербург, 2020. – 448 с.: ил.
6. Прохоренок Н.А. Python 3 и PyQt 5. Разработка приложений. – 2-е изд., перераб. и доп. / Н.А. Прохоренок, В.А. Дронов. – СПб.: БХВ-Петербург, 2019. – 832 с.: ил.
7. HTML, XHTML и CSS. Библия пользователя. 3-е изд. / Б. Пфаффенбергер, С. Шафер, Ч. Уайт, Б. Кароу. – Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2007. – 752 с.: ил.
8. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. 2-е изд. / Б. Хоган – СПб.: Питер, 2014. – 320 с.: ил.

Оглавление

	№ страницы
Введение	2
Глава 1. Разработка раскрывающихся меню	3
1.1. Разработка верхнего раскрывающегося меню	4
1.2. Разработка левого меню-аккордеона	11
1.3. Дополнительные функции для обработки меню	16
Глава 2. Разработка системы подключения дополнительных переводов	22
Глава 3. Разработка системы поиска по текстам на сайте	30
Глава 4. Разработка системы обратной связи с посетителями сайта	41
Заключение	53
Список использованной литературы	54
Приложение. Исходный код сайта (в отдельном архивном файле)	