

# СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”

Факултет по математика и информатика

Информационни технологии

## ДИПЛОМНА РАБОТА

Тема:

**WEB БАЗИРАНО ПРЕДСТАВЯНЕ НА ТРАДИЦИИТЕ  
И ОБИЧАИТЕ НА БЪЛГАРИЯ**

Дипломант: Камелия Минчева Косева

Специалност: Информатика

Специализация: Изкуствен Интелект

факултетен.№ 42360

Научен ръководител:

ст. ас. Евгения Ковачева

## **WEB БАЗИРАНО ПРЕДСТАВЯНЕ НА ТРАДИЦИИТЕ И ОБИЧАИТЕ НА БЪЛГАРИЯ**

1. Увод.....	3
2. Цел и обзор на дипломната работа.....	5
2.1. Цел на дипломната работа.....	5
2.2. Обзор на проблемната област.....	6
2.2.1. Литературни източници.....	7
2.2.2. Web източници.....	9
3. Проектиране на сайта.....	18
3.1. Структура на сайта.....	18
3.2. Проектиране на сайта.....	20
3.2.1 Изчерпателност и пълнота на информацията.....	22
3.2.2 Дизайн на страниците.....	23
3.2.3 Навигация.....	25
3.3. Проектиране на форума.....	29
4. Използван софтуер и реализация на функционалността.....	30
4.1. Използван софтуер.....	30
4.1.1 HTML.....	30
4.1.1.1. Проектиране на шаблоните на форума.....	34
4.2. Каскадни стилове.....	37
4.3. JavaScript.....	41
4.3.1. Обектите в JavaScript.....	44
4.3.1.1. Обектът navigator.....	48
4.3.1.2. Обектът window.....	49
4.3.1.3. Обектът document.....	51
4.3.1.4. Обектът images.....	53
4.4. PHP.....	59
5. Използвани инструментални средства.....	60
5.1. Adobe PhotoShop.....	60
6. Тестване, оценка и усъвършенстване на сайта.....	62
7. Заключение.....	64
8. Използвана литература.....	65
9. Приложение.....	66

## **1.Увод**

С развитието на информационните технологии през последните десетилетия компютърът зае своето неотменно място в обществото. Навлязоха нови гъвкави методи за визуализиране и обработка на използваната информация. Интернет предоставя възможност да се получава бърз, достъп до данни, даващи пълна информация на клиента, без ограничения във времето. Това би спестило досадното лутане между справочници, помагала, учебници. Днес Web-сайтове се очаква да съдържат много информация и да имат добро представяне, но не и да разсейват с труден и тромав интерфейс. Информацията в него трябва да е лесно откриваема и актуална. Един ясен и динамичен Web сайт е голяма ценност за потребителя, който иска бързо и лесно да достигне до желаната информация.

Целта на настоящата дипломна работа е да се разработи Web сайт, предлагащ изчерпателна информация за автентичните българските обичаи и традиции, така, както са били чествани през миналите столетия до средата на XX век. Тъй като интересът на чуждестранните граждани към страната ни нараства значително през последните години, основна цел на дипломната работа е този сайт да бъде достъпен и за тях. По тази причина сайтът е разработен както на български, така и на английски език.

Дипломната работа може условно да бъде разделена на две части – динамична и статична. Сайтът е разработен от екип от две дипломантки и по тази причина описанието на разработката е разделено на две части:

- статична част, обхващаща съдържанието на WEB-сайта, обзор на проблемната област, оформление, структурата и йерархията;
- динамична част, реализираща взаимодействието с потребителите – проектирането и разработването на форум, даващ възможност на потенциалните читатели на сайта да обменят мнения.

Тъй като дипломната работа е разработена в екип с Надежда Каменова Иванова, всеки един от тези етапи ще бъде разгледан в настоящата дипломна

работа, но по-подробно ще бъде описан само от една от двете дипломантки. В настоящата дипломна работа повече внимание ще бъде обърнато на статичната част на сайта и на неговото дизайнерско оформление. Взаимодействието с потребителите, т.е динамичната част от разработката ще бъде разгледана с повече подробности в дипломната работа на Надежда Каменова Иванова.

## **2.Цел и обзор на дипломната работа**

### **2.1.Цел на дипломната работа**

Настоящата дипломна работа има за цел разработването на Web сайт, съдържащ информация за традициите в празнуването на някогашните обичаи и обреди на българите. Това включва по-значимите църковни и народни празници, които са разделени според народния български календар. Предназначението на тази дипломна работа е да обхване колкото е възможно повече материали по темата, да ги систематизира и да създаде Web сайт, който да е атрактивен за потребителите.

Информацията в сайта трябва да е поднесена по не ангажиращ и увлекателен начин, достъпна за широк кръг от потребители. Обичаите трябва да бъдат представени както според църковния календар, така и според езическите вярвания на българите. Текстът следва да е богат в стилистично отношение и да бъде поднесен на български и английски език, тъй като стремежът е сайта да спечели не само български потребители, но и такива от цял свят. Поднасянето на графична информация би допълнило описанието на българските традиции. Трябва да има място на сайта, на което потребителите да могат да обменят информация, а именно форум.

Преди да започнем работа по нашето приложение бе необходимо да го разделим на няколко основни компоненти и да определим как ще работи всеки един от тях, и как те ще се съвместят заедно.

Въз основа на поставената по –горе цел се оформиха следните задачи:

- Изследване на източниците на информация за българските обичаи и традиции, въз основа, на които ще бъде построен сайта. Това включва проучване на наличната информация в Интернет пространството с подобно съдържание, както и на съществуващите книжни източници, описващи народните обичаи и традиции;

- Проектиране на решението. Какви са целите на сайта и как трябва да бъде организиран?
- Избор на софтуер за разработка на сайта.
- Проектиране на форума. Проектиране на базата данни, която ще обслужва форума.
- Реализация на графичния интерфейс на сайта. Подредба на менютата, графичните ресурси, форума и статичната информация.
- На базата на по –горните задачи да се създаде окончателният вариант на сайта.

В настоящата дипломна работа ще бъде обърнато по-голямо внимание на разработката на статичната част от WEB-сайта – на изборът на информацията, която да се съдържа в него, на източниците, използвани в разработването му, както и на графичното оформление. Останалите етапи от проектирането и направата на сайта ще бъдат разгледани в дипломната работа на Надежда Каменова Иванова.

## **2.2.Обзор на проблемната област**

За създаването на дипломната работа се наложи да бъдат изследвани голям брой източници, който послужиха за основа на сайта. Условно изследователският процес може да бъде разделен на няколко части:

- Изследване на възможно най-голям брой литературни източници, отнасящи се до народните обичаи и традиции на България;
- Проучване на Интернет пространството с цел разглеждане и оценяване на съществуващите сайтове, на чиято основа да бъде реализиран настоящия сайт, разчитайки на това настоящата работа да има съществени подобрения;

Поради невъзможността в настоящия сайт да бъде включена цялата намерена информация по темата се наложи наличната такава да бъде

подредена и сортирана, като стремежът е тя да остане максимално пълна, точна и изчерпателна, даваща възможно най-много подробности за народните обичаи и традиции.

### **2.2.1. Литературни източници**

Една от най-големите ценности на българите са били и винаги ще бъдат семейството и българските традиции. За народните празници и обичаи са написани много книги и ще продължават да се пишат още такива. Интересът към духовното богатство на българина няма да изчезне никога. Още повече сега, когато е необходимо преосмисляне и връщане към българските обичаи и традиции, чрез които прадедите ни предават своята сила и мъдрост и засилват националното ни самочувствие. Книгите посветени на тази тема са толкова много, че не би било възможно за един живот да бъдат прочетени всички. За създаването на настоящия сайт бяха прегледани много от тях, но като основа на настоящата разработка най-пълноценно бяха използвани само част от тях.

В основата на текста за българските обичаи и традиции стои книгата на Димитър Маринов – “Религиозни обичаи на българите” (1906г.). Книгата съдържа много задълбочено изследване на народните обичаи и обреди от началото на миналия век. Описани са подробно не само най-големите празници, чествани от българина, но и много от по-слабо известните, също толкова интересни и пленяващи обичаи, за които трудно се намират сведения и почти не се споменава в други източници. За всеки от празниците е описан както начина, по който той се празнува в цялата страна, така и специфичните за някои области обреди, характерни само за дадена част на България. Интересно е да отбележим, че в книгата се споменават не само обредите, характерни за даден празник, но и обредната трапеза, ястията които също носят традицията в себе си. В книгата е поместен и богат снимков материал за някои от празниците.

Основна част от информация за обичаите поместена в разработения сайт бе събрана от горе описаният източник. Недостатък на тази книга е старият стил на езика, който е претърпял много промени от началото на миналия век до настоящия момент , което наложи преработването на информацията, за да се предаде по-съвременно звучене на предложените текстове.

Също голяма полза в процеса на събиране на информация за българските традиции и обичаи изигра книгата-енциклопедия „Календарни празници и обичаи на българите” (издадена от академично издателство „Марин Дринов”, през 2000г.). В нея може да се намери информация за повечето от обичаите, описани в споменатата по-горе книга, но представена в значително по-синтезиран и изчистен вид.

Поредната книга, която беше проучена и помогна в процесът на систематизиране на информацията за българските обичаи е книгата на Надя Петрова - „Празници на българина” с подзаглавие „Църковни и семейни; Традиции и обичаи; Трапезни” (издателство „Парнас”, 2001г.). Народните ни обреди и обичаи са описани подробно и по увлекателен начин. За част от празниците, характерни с богатата си обредна трапеза са поместени и рецепти, даващи полезна информация за начина на приготвяне на празничните ястия.

За някои от най-големите празници на българите са написани отделни книги, засягащи само съответния празник. Те също бяха от полза в процеса на събиране и отсяване на данните за даден обичаи. Някои от тези книги са:

- „Изкуство и народни обичаи” с автор Лазанка Йорданова( издателство „Народна просвета”, 1988г.);
- „Етнография на България”, Христо Вакарелов, 1977г.;
- Информация за празника „Еньовден” ни помогна да съберем книгата „Еньовден”, написана от Валентина Ганева-Райчева. В нея намерихме не само подробно описание на характерните за този ден обреди, но и богат снимков материал;



- Сведения за Коледа и обичаят сурвакане успяхме да открием и в написаната от Маргарита Василева книга „Коледа и сурва“;
- За едни от най-интересните ни народни обичаи кукерските и сурвакарските, намерихме нужната ни информация от книгата на Живка Стаменова – „Кукери и сурваки“;

Целта на настоящата дипломна работа е да събере и опише колкото се може по-подробно и изчерпателно народните ни обичаи и традиции. Надяваме се с помощта на споменатите книжни източници да сме постигнали успех с реализирането на тази цел. Сравнявайки написаното във всички тях се опитахме да отсеем най-интересното и най-характерното за всеки един от българските празници.

Една част от информацията за обичаите е събрана от Интернет източници, някои от които ще бъдат описани по долу.

Не е възможно в настоящата дипломна работа да се обхванат всички имена и произведения, написани по въпроса. Въз основа на трудовете на посочените по-горе автори, стана възможно създаването на голяма част от настоящия сайт.

### **2.2.2. Web източници**

В настоящата разработка на дипломната работа ще бъдат разгледани предимно българските сайтове, съдържащи информация за българските обичаи и традиции. Чуждестранните сайтове с подобно съдържание, повлияли при разработката на настоящия сайта ще бъдат по-обстойно разгледани в дипломната работа на Надежда Каменова Иванова.

По време на създаване на настоящата дипломна работа бяха разгледани много Web източници с подобно съдържание. Стремехът беше чрез изследването на тези сайтове не само да се допълни и сравни нужната информация, но също така и за да се определят условията (характеристиките), на които трябва да отговаря разработеният сайт. Основните белези, по които

бяха сравнявани и изследвани намерените сайтове, и които оказаха влияние върху окончателният проект на настоящата дипломна работа са:

- **Въведение** - разглеждане на заглавната част на даден сайт, колко атрактивна и увлекателна е тя, до колко може да задържи вниманието на потребителя.
- **Определяне потребителите на дадения сайт** – кой би посетил и разгледал дадения сайт.
- **Интерфейс на сайтът** - графичен изглед на сайта, до колко е атрактивен или претрупан, интуитивна ли е ориентацията по страниците.
- **Изчерпателност и организация на информацията** - каква информация съдържа дадения сайт и до колко тя е покрила засегнатата област и по какъв начин е структурирана.
- **Навигация** - до колко лесно и интуитивно става предвижването по страниците на сайта.
- **Скорост на зареждане** - с каква скорост се зарежда сайта, до колко са оптимизирани графичните ресурси.
- **Софтуер** - какъв софтуер е използван за изработването на сайта.
- **Динамичност на сайта** - има ли възможност потребителя да взаимодейства със сайта по някакъв начин.

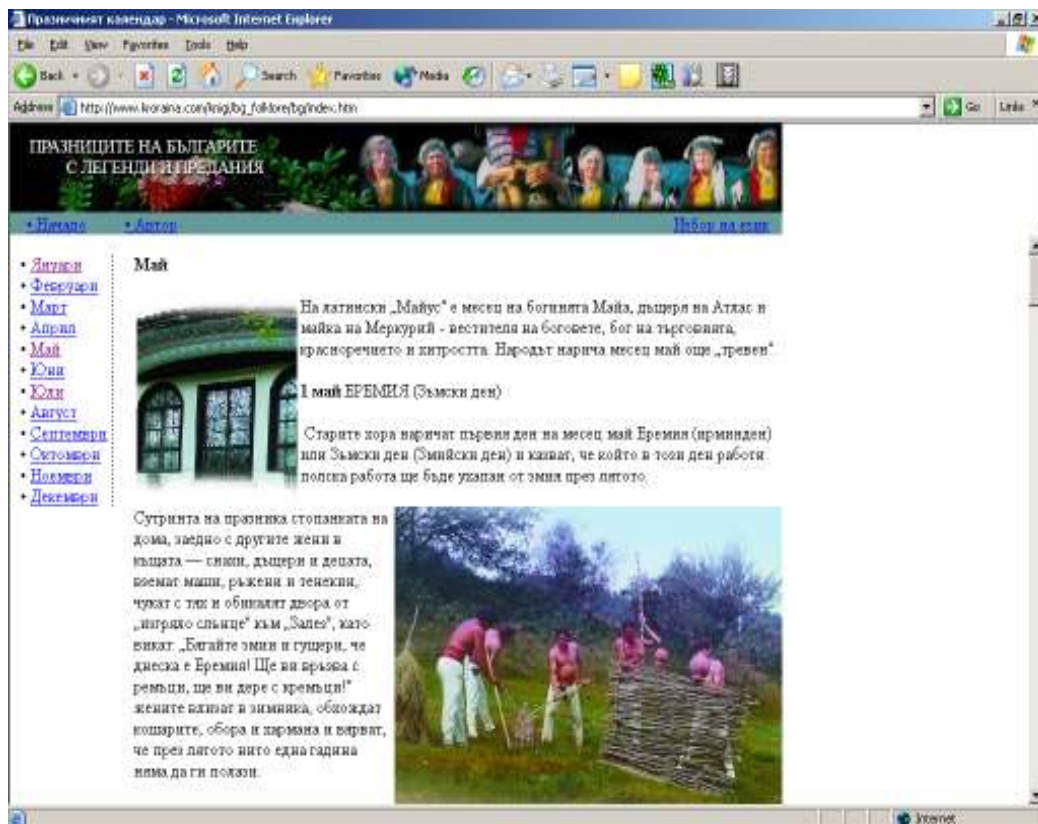
Някои от разгледаните сайтове съдържат много подробна информация за българските обичаи и традиции, докато други разполагат с бегло и непълно описание на честваните от българите празници. Наличните данни в повечето от изследваните сайтове са откъслечни и лошо подредени и систематизирани. В много от тях навигацията е лоша, трудно е връщането назад, а ако го има потребителят се обърква откъде точно е минал. Също така в повечето липсва възможност за търсене. Друг основен недостатък на по-голяма част от тях се състои във факта, че са изцяло на български език, което затруднява използването им от чужденци.

Направеното проучването на интернет пространството спомогна за набелязването на основните изисквания, на които да отговаря разглежданият сайт, за да даде максималното на потенциалните си потребители, както по отношение на информацията, така и по отношение на дизайнерското оформление на сайта.

Един от проучените сайтове е <http://www.viavita.net/>. Информацията за обичаите е много подробна и добре изложена, но основен недостатък на този сайт е, че в него са описани само една минимална част от българските празници и традиции. Друга негативна черта, която той притежава е лошата навигация – информацията за някои от обичаи се показва в същата страница(фрейм) , докато за други се отваря напълно самостоятелен прозорец, което до известна степен е неочаквано и объркващо за потребителя. Като недостатък смятаме също така и дизайнерското представяне на сайта - цветовата гама не е подбрана добре, снимковият материал е с лошо качество.

Още един от откритите сайтове, който оказва влияние върху окончателното решение за начина, по който да бъде разработен настоящия проект е <http://free.bol.bg/slavpagan/>. В началото на сайта се зарежда анимиран текст, който успява да привлече вниманието на читателите. Навигацията е почти интуитивна и потребителите лесно могат да се ориентират къде точно се намират в момента и как да отидат на мястото, което желаят. Голямо въздействие върху читателя оказват и добре подбраните цветове, които спомагат още повече за задържане вниманието му върху сайта. От този сайт се породих идеята за навигацията, реализирана в настоящата дипломна работа – падащи менюта, от които потребителят може да достигне желаното от него място и да открие необходимата му информация. Сайтът е преведен и на английски, което е една от предпоставките за неговия успех. Липсата на информация за църковните празници и наличието на подробно описание само на част от езическите обреди на българите може да се отчете като недостатък на сайта спрямо изискванията, на които би трябвало да отговаря сайт с подобно съдържание.

Друг сайт, съдържащ информация за народните ни обичаи е:  
[http://www.kroraina.com/knigi/bg\\_folklore/bg/index.htm](http://www.kroraina.com/knigi/bg_folklore/bg/index.htm).



Фигура 1

От гледна точка на пълнота и изчерпателност на информацията този сайт е един от най-добрите, намерени в българското интернет пространството. Той спомогна значително за набирането на нужните данни за написването на настоящата дипломна работа. Описанието на всеки един от празниците е подробно и задълбочено, но не се споменават църковните обичаи. За някои от по-големите празници е поместен и снимков материал. Наличната информация е преведена на няколко езика – английски, немски, френски и руски, което би привлякло много чуждестранни потребители и би увеличило значително интересът към този сайт.

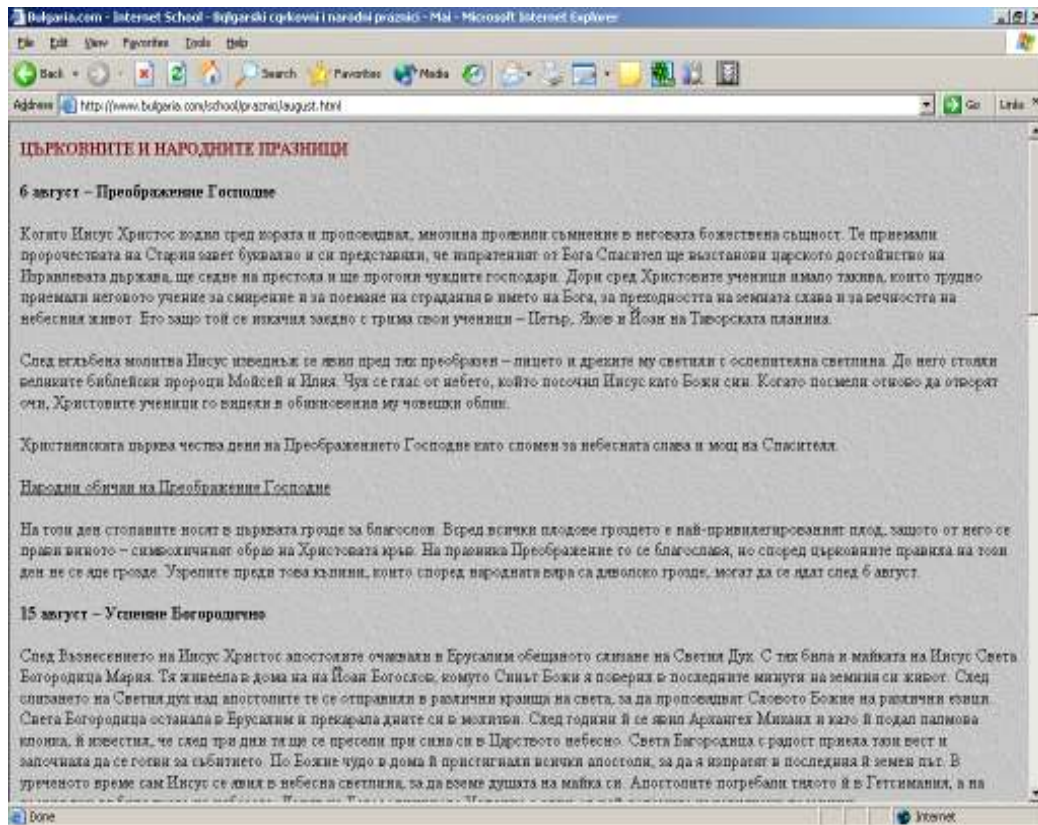
Като недостатък на този сайт може да се отбележи дизайнерското решение и лошата навигация – обичаите са структурирани по месеци, като

всички празници през даден месец са в един файл, което прави невъзможно самостоятелното разглеждане на точно определен от тях; за потребителя е много трудно да прочете описанието на исканият от него празник освен, ако не знае точната дата, на която той се чества. Но и в този случай намирането на желаната информация не е толкова лесно и интуитивно. Възможностите за придвижване в сайта са много малко и достъпни само на определени места.

Цветовата гама, използвана в сайта е една от характеристиките, които не правят добро впечатление на потребителите – избраните цветове не си подхождат и като цяло не правят добро впечатление.

С все по-бързо развиващите се технологии динамичността на сайтове и взаимодействието с потребителите се смята за една препоръчителна характеристика, която всеки сайт би трябвало да притежава. Затова статичността на разглеждания сайт може да се сметне за негов недостатък, тъй като на потребителя не е предоставена възможност да оказва влияние върху сайта.

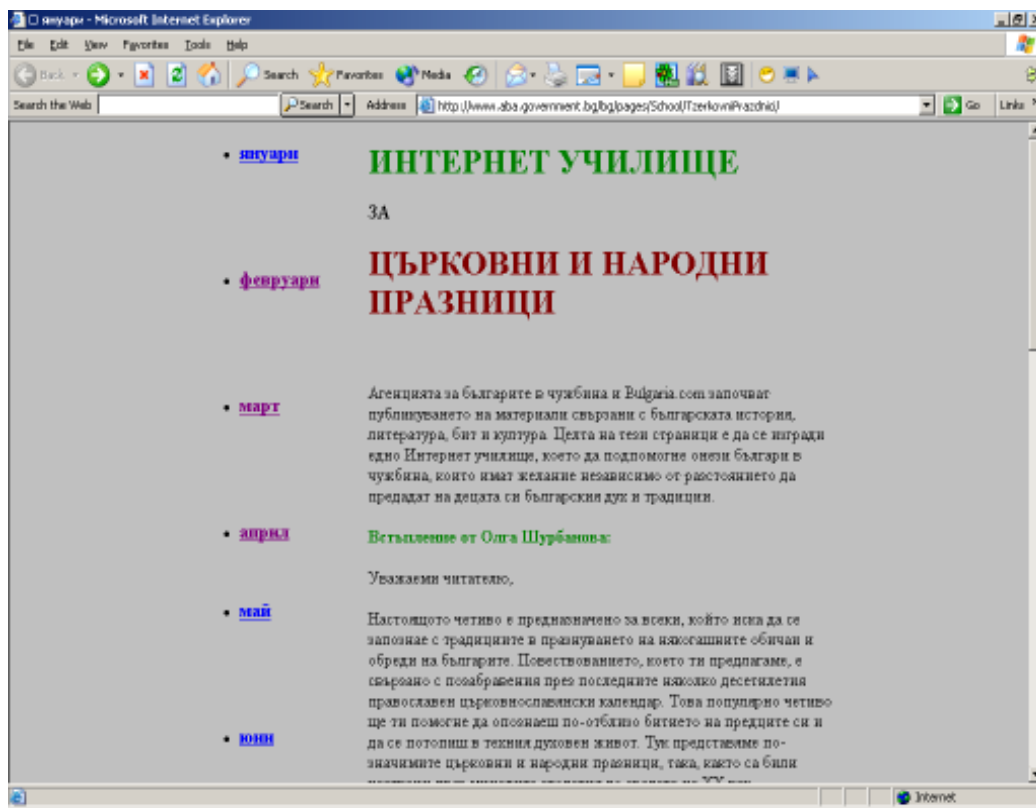
Подобен на предишния разгледан сайт е <http://www.bulgaria.com/school/praznici/index.html>. В него също е предложено по-обширно описание на обичаите, но за разлика от гореспоменатият сайт тук са представени много по-малко на брой празници и обичаи. Тази разработка е една от малкото, съществуващи и българското интернет пространство, в които се описват и църковните празници на българите.



Фигура 2

От гледна точка на дизайнерското представяне на сайта могат да се открият същите недостатъци – лоша навигация, неподходящ подбор на цветовете и статичност.

В следващия от проучените сайтове (<http://www.aba.government.bg/bg/pages/School/TzerkovniPrazdnici/>) са описани подробно голяма част от църковните празници на българите. За някои от по-интересните обичаи е добавен и снимков материал. По отношение на пълнотата на предоставената информация, този сайт е един от най-подробните в българското интернет пространство, спомогнал за събирането на необходимата информация за разработката на настоящата дипломна работа:



Подобно на голяма част от сайтовете, описващи българските традиции и в този сайт празниците са разделени по месеци, като за всеки месец подробно са описани голяма част от най-значимите църковни и народни обичаи на българина, разделение което е използвано и в настоящият сайт. Информацията в сайта е предложена само на български език, което е един от неговите основни недостатъци.

Както и при по-голяма част от гореспоменатите сайтове и тук дизайнерското оформление на страниците поставя тази разработка сред не толкова атрактивните сайтове, съдържащи подобна информация. Навигацията на сайта е една от характеристиките, които поставяме сред определящите за потребителския интерес към даден сайт, и която в разглеждания сайт повече затруднява потребителя, отколкото го улеснява. И в този сайт, както и в някои от споменатите по-горе сайтове, информацията за празниците през даден месец е поместена в един файл, което прави невъзможно разглеждането на даден празник самостоятелно. Липсват връзки между взаимосвързаните

обичаи, което би улеснило читателите в намирането на търсената от тях информация. Невъзможно е да се намери описанието на даден празник, ако не е известна датата, на която той се чества, но даже и така за потребителя е трудно и досадно да достигне до интересуващата го част, тъй като файловете са големи, липсват бутони, улесняващи придвижването в сайта и потребителят е принуден да търси продължително, за да открие интересуващите го сведения.

Сайтът <http://www.librousse.mlnk.net> е още един от проучените, който оказва влияние при проектирането на настоящата дипломна работа . В него също може да се намери информация за много голяма част от народните ни обичаи и традиции, но в списъка липсват някои от по-малко известните празници. Налегнато е предимно на църковната страна на празниците. И тук информацията е единствено на български език, което не би позволило на чуждестранния потребител да научи нещо за обичаите и традициите на българите.





В интернет пространството съществуват още много български сайтове, съдържащи подобна информация, но в тях се описват само малка част от обичаите ни, или само един конкретен и в повечето случаи темата не е засегната достатъчно задълбочено и подробно. По-голяма част от тези сайтове не са проектирани по лесен и удобен за потребителя начин – навигацията в тях е лоша, за читателите е трудно да открият нужната им информация, придвижването в тези сайтове е затруднено и е почти невъзможно за потребителя да проследи откъде е минал и как да се върне на мястото, от което е тръгнал.

Вземайки предвид всички тези негативни страни на съществуващите сайтове за обичаите в България, ние се постаряхме да създадем по-богат, с добра навигация и илюстриран с много изображения сайт за българските обичаи.

### **3.Проектиране на сайта**

#### **3.1.Структура на сайта**

След направеното проучване на различни български сайтове с подобно съдържание, достигнахме до извода, че в интернет пространството темата за българските традиции и обичаи не е засегната достатъчно подробно.

Въпреки, че сред намерените сайтове, засягащи тази тема имаше и такива, в които българските обичаи са описани подробно и обстойно, те все пак не представят всички обреди и традиции. В повечето от сайтовете може да се намери информация само за най-големите ни празници, а по-малко известните почти не се споменават.

Също така предлаганата информация, в повечето случаи, може да се намери предимно на български език, което е една от основните отрицателни характеристики на тези сайтове. Те биха били полезни само на българските читатели, желаещи да научат повече за традициите и обичаите ни, но един чужденец не би могъл да извлече нужните му данни от тези сайтове. Поради тази причина настоящата дипломна работа е разделена на две части – на български и на английски. Изборът на английски език не е случаен – той е сред най-разпространените езици и в днешно време голяма част от хората по света говорят този език.

Стремежът на настоящата разработка е да допринесе за запознаването на чужденците с българското духовно наследство, както и за обогатяване знанията на българските читатели. Сайтът е организиран по начин , който не затруднява читателите, навигацията е интуитивна и лесно разбираема за потребителите.

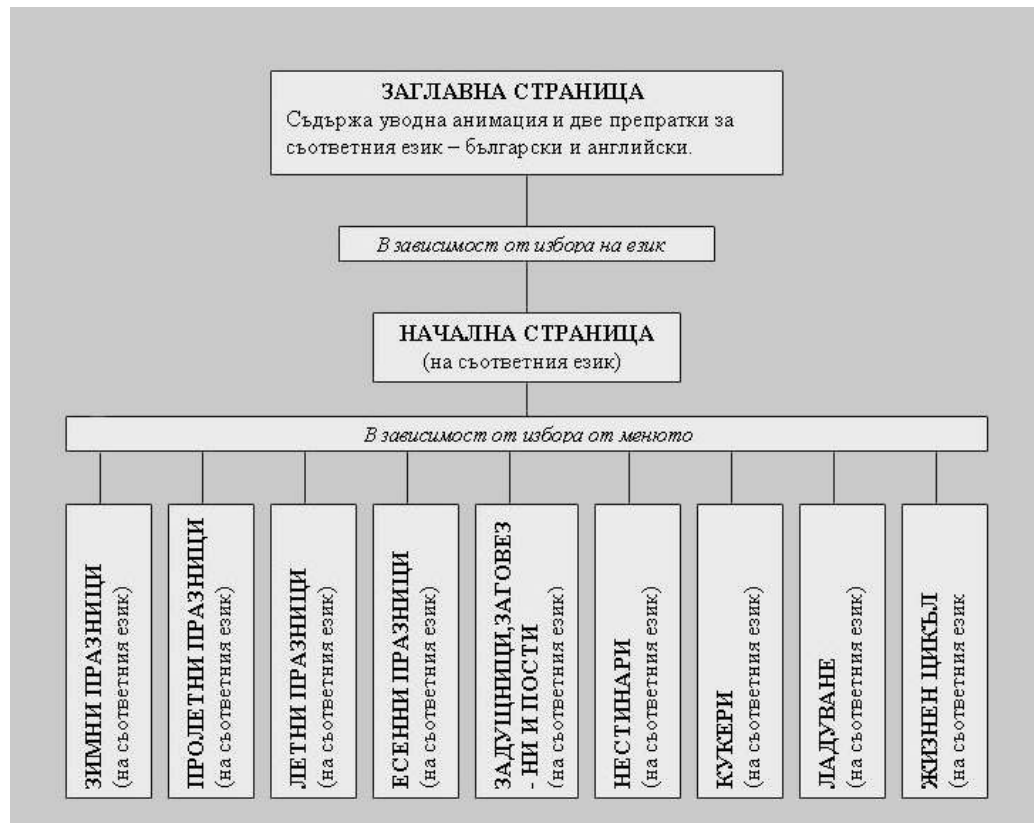
Въз основа на събраната от различните източници информация и начина и на представяне, празниците са групирани според българската етнография в няколко цикъла – зимен, пролетен, летен и есенен. Празниците във всеки един от циклите са разделени по месеци, както следва:

- зимен цикъл – декември, януари, февруари;

- пролетен – март, април, май и великден;
- летен – юни, юли, август;
- есенен – септември, октомври, ноември;

Някои от обичаите са обособени като самостоятелни елементи и не се включват в нито един от изброените раздели:

- заговезни, задушници, пости – тук са включени Сирни заговезни, Месни заговезни, Богородични заговезни, задушници, пости;
- кукери;
- ладуване;
- нестинари;
- жизнен цикъл – тук са описани обичаите, свързани с раждането на дете; кръщаването на дете; сватбените обичаите и погребалните обреди;



Фигура 3. Структура на сайта

### 3.2. Проектиране на сайта

На базата на направеното изследването на различните сайтове с подобно съдържание определихме основните характеристики и функционалността, на която да отговаря настоящата разработка. Най-важните от тях са:

- Дизайн на началната страница - заглавната страница на сайта да е достатъчно атрактивна, за да привлече вниманието на читателя още в самото начало, като в същото време не е тровава и претрупана.
- Дизайн на страниците, съдържащи основната информация за обичаите – изборът на цветовата гама в сайта е от голямо значение, тъй като цветовете също оказват влияние върху

потребителите и ако не са подходящо подбрани биха отблъснали голяма част от тях.

- Изчерпателност и пълнота на информацията – от направеното проучване достигнахме до извода, че в настоящият момент почти не съществуват web-сайтове за българските обичаи, в които да се съдържа описание както на езическите обреди, запазили се до наши дни, така и на църковните празници на българите. Затова стремежът на настоящата разработка е да се намери информация и максимално подробно да се опише всеки един от обичаите ни.
- Навигация – повечето от разгледаните съществуващи сайтове за българските обичаи не предлагат добро придвижване и лесен начин за намиране на желаната информация. Затова от голямо значение е в настоящата разработка този проблем да е решен по възможно най-добрият начин, който да осигурява интуитивен начин за придвижване на потенциалните потребители в рамките на сайта.
- Динамичност на сайта – един изцяло статичен сайт не би привлякъл толкова много потребители, колко някои динамичен такъв, подобряващ навигацията в него, както и даващ възможност на потребителя да напише своето мнение или коментар на дадената тема.
- Скоростта на зареждане също не е маловажна, защото всеки потребител би се отказал да разгледа един сайт, който се зарежда прекалено бавно и трябва да се изкачва дълго време, за да се обнови информацията в него.

### **3.2.1 Изчерпателност и пълнота на информацията**

В списъка с народните обичаи и традиции на българите са включени както най-големите български празници, така и някои по-малко известни обичаи, които за позабравени в наши дни, но които са имали голямо значение за нашите предци и крият богата обредност, имаща предимно езически характер. Следващият етап от изследователския процес бе да се раздели събраната от различните източници информация, необходима за разработката на дипломната работа на части, подходящи за представяне в сайта.

За всеки един от празниците е предоставена максимално пълна и точна информация – описани са както самите обреди, характерни за празника и начина им на честване в различните области на страната, така и църковния обичай, свързан с него, а също и обредната трапеза, приготвяна на този ден.

Тъй като информацията, която съдържа настоящият сайт няма да бъде променяна или допълвана, решението, което взехме бе сайтът да е статичен. Въпреки, че потребителите не могат да променят съдържанието на даден документ, те все пак имат възможност да напишат своят коментар или предоставят допълнителна информация за някои от обичаите в проектираният към сайта форум.

Проектирането, разработката и функционалността на форума ще бъдат описани в повече подробности в дипломната работа на Надежда Каменова Иванова.

### **3.2.2 Дизайн на страниците**

С цел улеснение на потребителя и по-лесна навигация в сайта, всеки един от празниците и свързаните с него обичаи са описани в самостоятелен документ. Поради същата причина страниците в сайта са разделени на няколко типа:

- заглавна страница – това е страницата, зареждаща се в началото на сайта;
- информационна страница – в този тип страници се съдържа основната информация за съответния празник;
- управляваща страница – в нея е даден списък на празниците през даден месец по дати;
- картина галерия – в страница от този вид е поместен снимков материал за съответния празник, който е разделен от самият текст, описващ даденият обичай. Това е направено с цел да се избегне претрупването на информационните страници със снимков материал, което би довело до нарушаване баланса между графичната и текстовата информация: Една страница със снимков материал изглежда по следния начин:



Фигура 4

При натискането с бутон на мишката върху дадено изображение то се зарежда страницата, като по този начин се предоставя възможност на потребителя да разгледа по-добре исканата снимка, чийто размер е по-голям.

За да бъде сайтът лесен за ориентиране и за да се избегне объркването от прехвърлянето от една страница в друга, към всяка от страниците е добавен заглавен текст. Във всеки един момент потребителят знае къде точно се намира и откъде е минал, за да стигне до желаната информация.

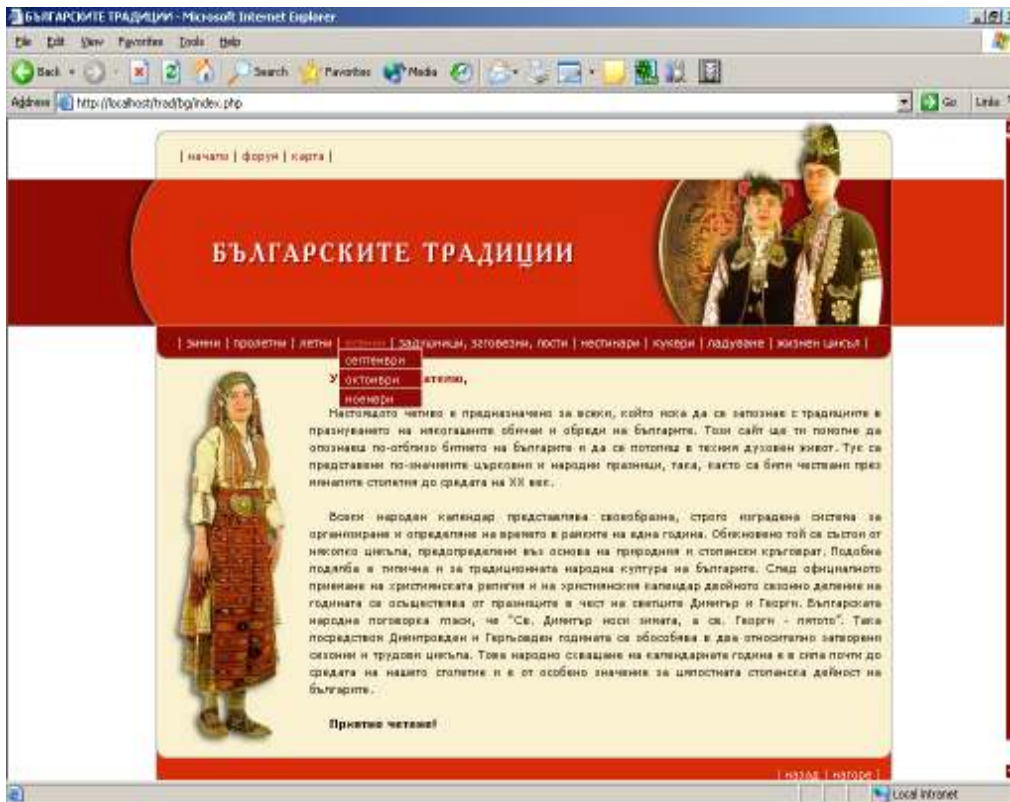
С цел постигане на еднообразен дизайн на сайта, е установен единен стил на поднасяне на информацията, местоположението на връзките, цветовата гама и др. По този начин потребителя се адаптира по-лесно към дизайна, разположението на връзките и бутоните в страницата, използвани цветове на навигационните елементи.



От голяма важност е и изборът на цветовете, които ще бъдат използвани, тъй като при първоначален поглед върху дадена страница, потребителя забелязва първо съвместимостта между цветовете и графиките . В настоящият сайт се стремим да използваме не натрапчиви, меки цветове, които да не дразнят окото, но в същото време са контрастни, за да може потребителят по-лесно да разграничи информационната част от навигационните елементи в страницата.

### **3.2.3 Навигация**

При стартирането на сайта се зарежда заглавната страница, в която е използван Flash за създаването на анимация, чиято цел е да привлече вниманието на потребителя и задържи неговият интерес. От тази страница читателят може да избере езика, на който иска да му бъде представена информацията – български или английски. След като читателят вече е определил езика, се зарежда входящата страница на този език, която съдържа кратко въведение, информиращо читателя какво може да очаква и намери в настоящият сайт. От нея, с помощта на главното меню, читателят може да избере и прочете данните за интересуващия го обичай.

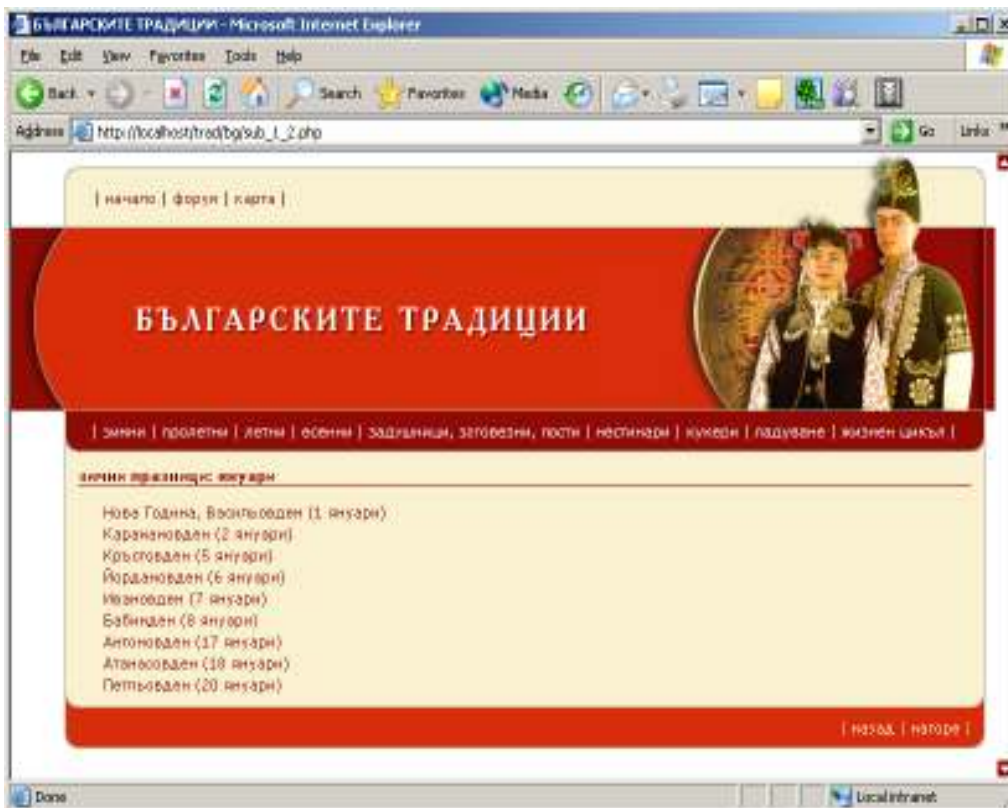


Фигура 5

Главното меню се състои от съответните бутони за избор на конкретна под тема:

- "Зимни",
- "Пролетни",
- "Летни",
- "Есенни",
- „Задушници, заговезни, пости”,
- „Нестинари”,
- „Кукери”,
- „Ладуване” и
- „Жизнен цикъл.

При избор на категория от менюто се зарежда съответната управляваща страница, съдържаща списък с празниците, чествани през дадения месец или съответната подкатегория на избраната категория.



Фигура 6

В началото на всяка страница е указано къде се намира текущата страница в структурата на сайта. Примерно, ако се намираме в зимни празници, декември месец и разглеждаме информацията предоставена за празника Никулден, тогава в горния край на екрана, веднага под менюто е указано: **зимни празници: декември: Никулден - 6 декември.**

Главното меню е видимо във всяка една от различните видове страници с изключение на заглавната.

Друга обща черта, която притежават всички страници са бутоните „назад” и „нагоре”, видими в долната част на страницата. Натискайки бутонът „нагоре”, читателят се „връща” в горната част на текущият документ, откъдето отново може да избере някое от подменютата. Бутонът „назад” зарежда последната посетена страница.

Следващата обща характеристика на страниците са бутоните, разположени над заглавния надпис („Българските традиции”):

- „Начало” – с натискането на този бутон се зарежда отново заглавната страница, откъдето можем да направим избор на език;
- „Форум” – този бутон служи за отварянето на нов прозорец, откъдето потребителят попада във форума към сайта и може да напише своят коментар по дадена тема или да започне нова дискусия;
- „Карта” – този бутон пренасочва читателя към управляваща страница, подобна на тези, които се зареждат при избора на даден елемент от главното меню, с тази разлика, че тук потребителят ще види списък с всички описани в сайта обичаи и традиции. Това е направено с цел улеснение и спестяване на време на потребителя;

Поредният елемент от навигацията на сайта са връзките между отделните информационни страници. При описанието на даден обичаи много често се споменават някои от останалите празници, имащи отношение към него. Името на всеки от споменатите обичаи е хипервръзка, която ни препраща в информационната страница на този обичаи. По този начин само с едно натискане на бутона на мишката върху името на празника зареждаме неговата страница.

В сайтът няма страници без изход, което предотвратява блокирането на потребителя на някое ниво от йерархията на сайта, тъй като главното меню е достъпно от всяка страница на сайта и по този начин предвижването по страниците става лесно и интуитивно.

Изглед на горе описаните страници може да откриете в Приложението на настоящата дипломна работа.

### 3.3. Проектиране на форума

Въпреки, че съдържанието на форума се генерира изцяло динамично в зависимост от това кой е потребителят, който разглежда форума, коя тема е избрана за разглеждане и т.н., външния вид на страниците изглежда по един и същ начин, независимо от информацията показана в нея. Когато потребителят реши да разгледа дадена тема, той ще види винаги една и съща страница, независимо от това, че текста в нея е различен – разположението на бутоните и информацията ще бъде еднакво в случаите когато потребителя разглежда мнение, дадено по една тема, както и кога реши да прочете мнение, дадено по съвсем различна тема.

Това е причината в разработката на форума съдържанието да бъде разделено от външното представяне на информацията. Необходимата информация се извлича от база данни с помощта на PHP функции, докато HTML кода, необходим за генерирането на страниците, които ще види потребителя е напълно отделено от тях. По този начин за показване на извлечената информация се използват няколко основни HTML файла, които ще наричаме шаблони, които определят начина, по който тази информация ще се визуализира в брауъра. Създадени са няколко основни HTML файла, които се попълват с нужната информация, извлечена от базата данни. Някои от тези шаблони са:

- user\_datafrom.html – това е шаблонът за регистрация на потребител;
- user\_login\_form.html – шаблон за влизане в даден форум;
- main\_post\_from.html – шаблон, показващ текстово поле, в което потребителят може да напише своето мнение;
- main\_topics.html – шаблон за разглеждане на дадена тема и т.н;

В настоящата дипломна работа ще бъде описан начина на създаване на HTML шаблоните, използвани във форума на сайта, а начина на извличане и предаване на нужната информация на HTML файловете, посредством PHP

функции ще бъде разгледан подробно в дипломната разработка на Надежда Каменова Иванова.

## **4.Използван софтуер и реализация на функционалността.**

### **4.1.Използван софтуер**

#### **4.1.1HTML**

За създаването на настоящият сайт бе използван езикът HTML(Hyper Text Mark-up Language). Това е универсалният език за създаване на web-страници, с чиято помощ може да се форматира текст, да се добавят графики, звук. Той е прост механизъм с вградени функции за форматиране и свързване на документи, за работа с таблици, фонове изображения и др. Страниците разработени с помощта на HTML като цяло са статични.

Едно от най-големите преимущества на HTML е неговата лесна реализация. Всичко, от което се нуждаем, за да създадем един HTML документ се съдържа в таговете, чрез които браузърът идентифицира документа и различните му части.

Таговете в HTML се разделят на две групи - обграждащи и разделящи.

Всички обграждащи тагове следват установена форма – начало и край:

- започват с лява ъглова скоба (<), име на таг, дясна ъглова скоба (>). Например: <HEAD>
- завършват с лява ъглова скоба (<), наклонена черта, име на таг, дясна ъглова скоба (>) - </HEAD>

Задължителните елементи на един HTML документа са:

- Тагове < HTML >, </ HTML > - те дефинират самият HTML документ; между тях се поставя всичко, което искаме да съдържа този документ;
- Тагове < HEAD >, </ HEAD > - дефинират частта от документа, която ще се използва от браузърите;

- Тагове < TITLE >, </ TITLE > - служи за идентификация на документа в уеб; текстът поставен между тези два тага се появява в заглавната лента на прозореца
- Тагове < BODY >, </ BODY > - между двата тага се дефинира частта от документа, която е видима в брауъра.

Разделящите тагове също започват с лява ъглова скоба(<), име на таг, дясна ъглова скоба(>) : (<P>), но за разлика от обграждащите тагове при тях не се използват затварящи такива. Използвайки този вид тагове можем да добавим нов ред, изображения, хоризонтални линии към форматирания текст. Например тагът <HR> добавя хоризонтална линия; <BR> служи за вмъкване на нов ред.

Основната структура на един HTML документ е следната:

```
<HTML>
<HEAD>
<TITLE>Български обичаи</TITLE>
<META http-equiv="Content-Type" content="text/html;
charset=windows-1251">
</HEAD>
<BODY>
Тук трябва да поставим текста, който искаме да се покаже в
брауъра
</BODY>
</HTML>
```

За създаването на един HTML документ се нуждаем само от един обикновен текстов редактор, в който може да създадем файл с разширение .html и брауър, в който да визуализираме създадения файл.

Най-разпространената кодова таблица за кирилица е *windows-1251*. Ако се зададе *windows-1251*, може да се пише текст на български в WEB страница и е сигурно, че брауъра ще интерпретира правилно шрифта, за да е възможно нормалното четене. Поради тази причина във всеки един документ от настоящата разработка е указана тази кодова таблица. Ако не се зададе нищо, по подразбиране кодирането е за латиница.

За да направим текста, който искаме да се покаже в браузъра малко по-подреден, прегледен и форматиран, в HTML документа може да използваме така наречените таблици.

Основните тагове, които се използват при създаването на една таблица са:

- `<TABLE>`, `</TABLE>` - двата тага указват началото и края на една таблица;
- `<TR>`, `</TR>` - с помощта на тези два тага определяме началото и края на ред от дефинирана таблица;
- `<TD>`, `</TD>` - поместеното между двата тага определя една колона в таблицата.

Можем да манипулираме таблиците по много начини – да добавим празно пространство, подравняване, цвят и много други тагове. Големината на дадена таблица може да бъде променен чрез увеличаване броя на редовете или на клетките от данни. Имаме възможност да определим заглавни редове, да накараме клетките от данни да обхванат повече от един ред или колона, да поставим фон, рамка и др.

Една примерна таблица изглежда по следния начин:

```
<table width="778" border="0" align="center" cellpadding="0" cellspacing="0">
.....
<tr>
  <td colspan="5">
</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td></td>
  <td colspan="2" bgcolor="#D92D0A" class="navt"><div align="right">| <a href="javascript:history.go(-1);" class="nav1">назад</a>
```



```
|<a href="#top" class="nav1">нагоре</a>|</div>
|</td>
|<td></td>
|</tr>
|<tr>
|<td colspan="5">
|</td>
|</tr>
|</table>
```

Един от атрибутите на таблицата е *border* – чрез него определяме дебелината на външната рамката на таблицата. В нашият сайт таблицата е „невидима” и рамката ѝ е с дебелина нула пиксела.

Атрибутите *width* и *height* определят съответно широчината и дължината на таблицата и нейните клетки. Стойностите могат да бъдат в пиксели или проценти. Атрибутът *padding* определя разстоянието между съдържанието на клетките и техните очертания.

Едно от най-често използваните свойства на HTML документ е създаването на хипертекстова връзка. Чрез хипертекстовия таг текущата страница свързва текст и/или изображение с други документи или части на същия документ с използването на котви:

```
<a href="#top" class="nav1">нагоре</a>
```

Хипертекстовия таг за котва е `<A>`, а частта за обръщение на тага е `HREF = „файл”`. Обръщението на хипертекста съдържа път (относителен или абсолютен) към документа (файла), до който се опитваме да осъществим връзка.

Кодът на тази таблица е взет от файла **footer.inc**. Това е последният ред от всяка една страница в сайта. Тъй като стилът на всички страници е сходен, т.е. таблиците са организирани по един и същи начин, атрибутите на

таблицата, редовете и колоните, тяхната визуализация е дефинирана посредством CSS. Неговото създаване е описано в следващата част.

Също така поради големият брой на страниците в документа и повтарящата се структура, част от нея е изнесена в отделни файлове – header.inc, footer.inc, които съдържат повтарящата се информация. Включването на тези файлове в страниците на сайта ще бъде разгледано с повече подробности по-долу.

HTML ни предоставя възможност да добавим графика или изображение към нашата страница. За тази цел използваме следния синтаксис:

```
<IMG SRC = "Име на изображение">
```

Последният таг бе използван многократно в настоящият сайт, давайки ни възможност да добавим много снимков материал към описанието на българските обичаи, да го оразмерим и поставим на желаното от нас място в дадена страницата.

#### **4.1.1.1. Проектиране на шаблоните на форума**

В разработката на файловете-шаблони, използвани за визуализиране на информацията във форума, както и в останалата част от страниците на сайта са използвани таблици. Таблиците, както и останалите от основни HTML тагове са описани по-горе.

Следния код е част от шаблона за регистрация на потребител във форума:

```
... <form action="{ $indexphp}" name=regform method=post
class=formStyle>
<input type=hidden name=action value="{ $actionName}">
<table class=forums>
<tr><td class=caption4 colspan=2>{ $l_fillRegisterForm}
<span class=warning>* -
{ $l_necessaryFields}</span><br>{ $warning}</td></tr>...
```

В този файл, както и в много от останалите шаблони, използвани във форума са използвани формуляри. Формулярите служат за предаване на информация от брауъра към сървъра. Информацията може да бъде предавана по няколко начина – радио бутони, полета за отметка, текстови полета и т.н. Всички формуляри са разположени в HTML документи между таговете `<form>` и `</form>`. Атрибутите, които могат да бъдат използвани с този таг са:

- `action` – това е URL-адреса на сървърната програма, която ще обработи въведените във формуляра данни;
- `method` – предоставя метода за обработка на входните данни; възможни стойности са `get/post`; по подразбиране, ако не е указан метод за такъв се избира `get`. Методът `get` се предпочита при операции, които не променят състоянието на сървъра, докато метода `post` се използва предимно при операции , които променят състоянието на сървъра –добавяне или изтриване на записи от базата данни и т.н.
- `name` – име на формуляра и т.н.

Брауърът получава входна информация от потребителя като визуализира полета, които могат да бъдат редактирани. Тези полета се задават чрез HTML тага `<input type = key/value>`. Типовете на въведената информация могат да бъдат няколко вида:

- `text` – служи за създаване на текстово поле:  
...`<input type=text name=icq maxlength=15 size=15 value="{ $icq}" class=textForm>`...; използван е многократно в шаблоните;
- `password` – подобно на текстово поле, но информацията е скрита; използван е при шаблона за регистриране на потребителите и при влизането им във форума:  
...`<input type=password name=passwd maxlength=32 size=20 value="{ $passwd}" class=textForm>`...;
- `select` – създава скролиращи се списъци, от които може да се избере само една стойност. Този тип е използван

при регистрирането на потребителите, за да предложи избор на държава:

```
...<select          name=fromSelect          class=textForm
onChange="JavaScript:insertCountry();" style="width:200px;">
<OPTION VALUE=""></OPTION>
<OPTION VALUE="Afghanistan">Afghanistan</option>...;
```

○ hidden – предоставя двойка име/стойност без визуализиране на обекта на екрана:

```
... <input type=hidden name=action value="{ $actionName }">...;
```

○ submit ;

```
...<input          type=submit          value="{ $userTitle }"
class=inputButton>...;
```

○ textarea – текстово поле, което може да заема няколко реда;...

Съществуват още възможности за въвеждане на информация посредством типовете на формуляра, които също са използвани в този сайт, но не всички ще бъдат описани.

## 4.2.Каскадни стилове

С цел улеснение и унификация на стила в настоящата дипломна работа са използвани каскадни стилове.

Добавянето на каскадни стилове – CSS(Cascading Style Sheets) към HTML документ позволява да се подобри вида на дадена web страницата, както и да установи много по-голям контрол над визуализация на различните части. Чрез набори от стилове могат да контролират шрифта, цвета, размера на текста, както и разстоянията между отделните HTML елементи. Стилът представлява набор от (главно) визуални характеристики и може да се приложи към всеки HTML документ.

Каскадните стилове са първоначално разработени от World Wide Web консорциума и вградени в HTML 3.2 с цел да се разграничи съдържанието на един Web документ от неговия стил. Към всяка една страница може да се добави набор от каскадни стилове, който да определи и контролира стила ѝ.

Наименованието „каскаден” идва от начина, по който се прилагат дефинираните стилове. Той означава, че приложения върху един елемент стил влияе не само на конкретния елемент, но и върху всички негови наследници.

Един стил се състои от набор от свойства със зададени стойности във формата **свойство:стойност**. Като по този начин могат да се дефинират неограничен брой стилове:

```
font-size:12px;text-align:center.
```

Предимствата на каскадните стилове, поради които взехме решение да ги използваме са:

- Улеснява се създаването на общ формат за всички Web страници.
- Позволява управление на текста по начин, който е недостъпен за HTML. Може например да се настройват междуредията, цвета на фона и т.н.
- Спестяват време.

- Лесно се променят.
- Могат да се използват заедно с JavaScript за създаване на ефекти известни като DHTML.
- Промените във всяка страница се извършват от компютъра, а не от разработчика, което намалява възможността да бъде пропуснат някой елемент.

Наричаме правило(rule) декларацията на CSS стил за всеки отделен, а сборът от правила – таблица със стилове (style sheet). Името на тага се нарича селектор (selector) за правилото.

Съществуват три начина за вграждане на каскадните стилове в един HTML документ:

- вграден „inline“;
- вътрешен;
- външен;

В първия случай стилът може да се присвоява чрез атрибута *style* на елемента или чрез елемента <STYLE> и ще изглежда по следния начин:

```
<H1 style="font-size:12px;text-align:center;">Български обичаи</H1>
```

По този начин може лесно да се смени подразбиращото се представяне, само на определени елементи. Атрибутът *style* се прилага само върху елемента, към който е добавени, както и към неговите наследници, но не и към всички други елементи от същия тип.

Вторият начин на вграждане на каскадните стилове изглежда по следния начин

```
<STYLE>  
  H1 {font-size:12px;text-align:center;}  
</STYLE>
```

и позволява да асоциираме даден стил към определен тип елемент. Това гарантира, че при всяко появяване на тага <H1>, елементът ще бъде представен с помощта на дадения стил. Трябва да отбележим, че таблиците със стилове, съдържащи се в <STYLE> елемента, не се подчиняват на

каскадния ефект. Ако се дефинира едно правило за даден селектор в една таблица със стилове, а по-надолу дефинираме друго правило за същия селектор, то последното правило ще се използва в цялата страница.

За да се свърже един външен набор от стилове с даден HTML документ се използва елемента <LINK>:

```
<link href="../style.css" rel="stylesheet" type="text/css">
```

По този начин таблицата **style.css** се асоциира с текущата страница. Значението на различните атрибути е следното:

- атрибутът *rel* (relation - релация) указва, че връзката е от тип *stylesheet*;
- атрибутът *type* указва типа на свързания документ (*text/css*);
- *href* определя относителния или абсолютния URL адрес към документа, който се свързва.

С използването на последния начин на вграждане на каскадните стилове в HTML документа ние си осигуряваме използването на еднакъв стил за всеки един от елементите, дефинирани в CSS файла. Ако по някаква причина не искаме даден елемент да използва дефинираните стилове чрез атрибута *style* може да променим това и да се предефинираме елементът.

Първите два начина за дефиниране на каскадни стилове не са много практични, когато Web сайтът съдържа много страници, които искаме да изглеждат по един и същ начин в стилово отношение. Случаят с настоящата дипломна работа е точно такъв и това е причината в разработването на текущият сайт да използваме „външния” начин на вграждане на каскадни стилове. По този начин може да създадем цялостният изглед на сайта, използвайки една-единствена таблица със стилове, към която да се обръща всеки един от HTML документите ни.

Една от възможностите, които предлагат каскадните стилове е използването на класове. Те могат да бъдат използвани, когато искаме даден елемент да има различно държание в различните случаи. Класовете в CSS се дефинират по същият начин като обикновените правила за елементите, с тази разлика, че започват с точка, следвана от името на класа и правилата. Ето

един пример от CSS файла, използван в настоящата дипломна работа как може да дефинираме клас:

```
.contents{  
    color: #000000;  
    font-family: Verdana, Tahoma, Arial, Helvetica, sans-serif;  
    font-size: 11px;  
    font-weight: normal;  
    line-height: 16px;  
}
```

Извикването в HTML документа става по следния начин:

```
<td valign="top" class="contents">
```

С помощта на псевдо-класове може да се определи поведението на най-често използваните връзки и да се обособи в отделни класове. Недостатък на този тип дефиниране на каскадни стилове е зависимостта от браузъра – някои от съществуващите браузъри не поддържат част от свойствата на псевдо-класовете.

Псевдо-класовете започват с две точки и се дефинира в CSS, а не в структурата на HTML. Най-често използваните свойства в тези класове са:

- :link;
- :active;
- :visited....

Използват се за задаване цвета на връзките или други характеристики поддържани от браузъра.

Една от най-важните възможности на CSS (Куигли, 2004, стр.568) е тази за позициониране на обекти в страницата, оразмеряване на обектите и възможността те да се дефинират като видими или невидими – особеност, която е използвана за създаването на падащото меню в сайта. Това прави възможно преместването на обекти в различни части на страницата, преместването на текст и изображение, създаване на анимация и т.н. Обикновено при поставяне на тагове в HTML документ се започва отгоре



надолу. С помощта на CSS може да се зададе позиция на елемент, като елементите дори могат да се наслагват един върху друг.

Употребата на каскадни стилове в разработката на един web сайт е голямо улеснение и удобство за неговият проектант. С тяхна помощ може лесно да контролираме и уеднаквим вида и поведението на различните елементи в създадената HTML страница.

### **4.3.JavaScript**

С помощта на HTML можем да създаваме предимно статични сайтове – в тях можем да форматираме текст, да добавим изображения към документите ни, но това в общи линии е всичко, което той ни предлага . За да внесем енергия и жизненост на неподвижните web страници сме използвали скриптовият език JavaScript. Той позволява на иначе статичните страници да взаимодействат с потребителя и да реагират на събитията, настъпващи в страницата.

Една от основните причини да използваме JavaScript в разработката на сайта за българските обичаи е, че той се поддържа както от Netscape Navigator (от версия 2 нататък), така и от Internet Explorer (от версия 3 нататък).

JavaScript, първоначално известен като LiveScript, е разработен от Брендан Ейч в Netscape през 1995г. Той се стартира на повечето платформи и е хардуерно независим, което е една от основните причини да бъде използван за разработката на настоящата дипломна работа. JavaScript е обектно-ориентиран, интерпретаторен език, изграден директно в брауъра. Той е език за писане на скриптове от страна на клиента (clinet-side).

Тъй като JavaScript се асоциира с брауър, той до голяма степен се обединява с HTML. Но докато HTML се обработва от своя собствена мрежова библиотека , програмите на JavaScript се изпълняват от JavaScript интерпретатор, който обикновено е вграден в брауъра. Когато брауъра заяви

страница, съдържаща JavaScript, сървърът изпраща пълното съдържание на документа, включително конструкциите на HTML и JavaScript по мрежата до клиента. Когато страницата се зареди, HTML съдържанието се прочита и се показва ред по ред, докато стигне до отварящ таг на JavaScript и в този момент се включва интерпретаторът на JavaScript. Когато се достигне затварящия таг на JavaScript, се продължава с обработката на HTML кода.

JavaScript се използва за отчитане и реагиране на събития, породени от потребителя, като поставянето на курсора на мишката върху връзка или изображение. Подобренията, които той може да добави към една страница са:

- навигационни помагала (от навигационна гледна точка);
- скролиращи съобщения;
- диалогови прозорци;
- динамични изображения и др.

Използвайки HTML можем да създадем бутони, диалогови прозорци и формуляри за попълване, но не може самостоятелно да реагира на въведената от потребителя информация. Тази информация обикновено се комплектова от формуляра и се изпраща към сървъра за обработка. С помощта на JavaScript събитията, настъпили в следствие на потребителско действие могат да се обработят – да се провери за валидност въведената информация, да се реагира на натискане на бутон и др.

Нека проследим какво става, когато подадем заявка към сървъра за визуализиране на страница. Това са стъпките, през които преминаваме:

- въвеждаме URL в браузъра;
- браузърът изпраща HTTP заявка към подходящият сървър;
- сървърът връща съдържанието на web страницата от посочения URL;
- браузърът изпраща допълнителна заявка за всяка една от графичните страници;
- след получаване на достатъчно информация за определянето на необходимото място за графиките, браузърът показва страницата;

- графиките се появяват когато за тях се получи нужната информация.

Когато в един HTML документ използваме JavaScript този процес леко се променя. След като получим целият HTML документ, в стъпка 3 от по-горе изброените, се проверява за наличието на <script> тага. Ако има такъв таг, скрипта се изпълнява, но начина на изпълнение зависи от типа на скрипта (мястото му на дефиниране в HTML документа):

- Ако скриптът е дефиниран в <head> частта на документа, тогава той се игнорира, докато не се извика изрично;
- Ако скриптът е дефиниран директно в тялото на HTML документа, тогава неговият резултат ще бъде включен в страницата; по този начин той може да повлияе върху визуализацията на страницата;
- Ако скриптът представлява програма за обработка на дадено събитие, предизвикано от потребителя, той ще бъде изпълнен единствено в случай, че такова събитие наистина се случи.

Всеки един от изброените три начина за дефиниране на JavaScript в рамките на един HTML документ може да ни бъде полезен в една или друга ситуация - кой от тях ще използваме зависи единствено от нуждите на текущата страница.

В настоящата дипломна работа използваният най-често начин е вторият:

```
<head>....  
<script language="JavaScript1.2"  
src="quickmenu/sample_data.js"></script>  
...
```

Основните характеристики на езика Javascript се заключават в следното:

- Няколко основни типа данни:
  - Цифров тип, без ясно разграничаване между цели и реални;
  - Символен тип;
  - Логически тип;
  - Нулев тип;
- Слаба типизация в езика, една променлива може във всеки един момент да промени своя тип;
- Синтактична близост на езика с езика за програмиране C, почти същите оператори и инструкции;
- Езикът е основан на начален обектен модел, липсва понятието клас и не поддържа унаследяване;
- Няколко базови обекта за обработка на дати, таблици и математически пресмятания;

#### **4.3.1.Обектите в JavaScript**

JavaScript е базиран на прост обектно-ориентиран модел, т.е той поддържа обекти. Обектите са типове данни, които комбинират данни и функции за тяхната обработка. Всеки обект включва *свойства*, които описват характеристиките на обекта и *методи*, които обработват (манипулират) данните. Обектите могат да се подчиняват един на друг – всъщност всеки обект, подчинен на друг такъв, е също и свойство на този обект.

Един примерен обект е следния:

Нека обектът се казва Card и има следните свойства:

- name
- address
- work\_phone
- home\_phone

card1 е една инстанция на обекта Card. Можем да присвоим стойност и дефинираме свойствата на този обект по следния начин:

```
card1.name = "Sherlock Holmes";  
card1.address = "221B Baker Street";  
card1.work_phone = "555-2345";  
card1.home_phone = "555-3456";
```

Потребителят може да създаде свои обекти или да използва вградените в JavaScript такива. Всички обекти в JavaScript са наследници на обект, наречен Object.

За да създадем инстанция на даден обект можем да използваме така наречената функция конструктор:

```
function Card(name,address,work,home) {  
    this.name = name;  
    this.address = address;  
    this.work_phone = work;  
    this.home_phone = home;  
}
```

Когато функцията е извикана с ключовата дума new, тя играе ролята на конструктор и изгражда нов обект, а след това връща референция към него. Ключовата дума this се използва за обръщение към обекта, с който е извикана функцията.

Обект може да създадем и чрез използването на конструктора Object(). Резултатът от извикването на функцията-конструктор се присвоява на променлива, която съдържа указател към новия обект.

```
tom = new Card("Tom Jones", "123 Elm Street", "555-1234", "555-9876");
```

JavaScript предоставя няколко основни обекта, вградени в езика. Те управляват датата и часа, извършват математически изчисления, използват се за работа с низове, регулярни изрази, числа и други полезни единици.

Основните обекти са консистентни при различни имплементации и платформи и са стандартизирани от спецификацията ECMA Script10, което позволява на програмите да бъдат преносими и независими.

Тези обекти са:

- Array – представя набор от еднотипни елементи;
- String - чрез него могат да се манипулират символи и низове;
- Date – използва се за обработка на дата и час;
- Math – с него помощ могат да се обработват сложни математически операции. За да създадем инстанция на обекта Math не използване ключовата дума new, както е при останалите обекти;
- Navigator – съдържа информация за брауъра и неговите възможности;

JavaScript програмите са асоциирани с прозорец на брауър и документът се визуализира в прозореца. Прозорецът е брауърен обект, а документът е HTML обект. В брауърния обектен модел, прозорецът е главния обект, който се наследява от всички останали обекти:

- window
- navigator
- frames[]
- history
- document
- location
- screen

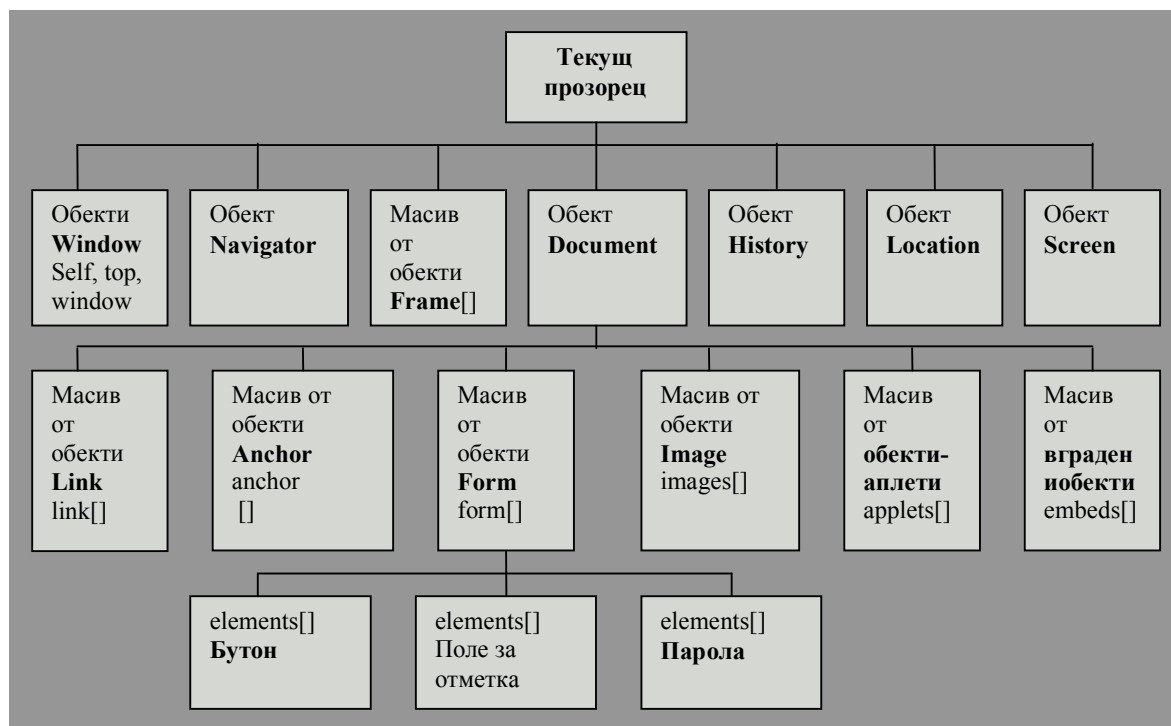
Документният обектен модел, наречен още DOM, се отнася за HTML документ и всички елементи и атрибути, асоциирани с него. JavaScript използва този модел, за да осъществи достъп до HTML елементите и атрибутите в рамките на дадена страница. Коренът на модела е документа.

Според World Wide Web Consortium(W3C) , документният обектен модел е платформено и езиково независим обектен модел, който позволява на програмите и скриптовете динамично да получават достъп и да обновяват

съдържанието, структурата и стила на документа (Куигли, 2004, стр. 279). Той имитира структурата на документа, за който се отнася – при работа с JavaScript DOM имитира HTML документа. Всеки елемент на HTML документа, като изображение, формуляр, връзка или бутон, може да се представи като JavaScript обект.

W3C дефинира първо ниво на DOM, създавайки индустриален стандарт за всички браузъри, напълно поддържан от Netscape 6 и Internet Explorer 5 и 6. Стандартът се състои от две части: първата определя начина на навигация и манипулиране с HTML и XML структура за основни обекти, свойства и методи ; втората дефинира строг набор от обекти, свързани с HTML.

Комбинирайки браузърния и документния обектни модели, JavaScript позволява да манипулираме всички елементи на страницата като обекти, от прозореца надолу по йерархията (фиг. 7)



Фигура 7 Комбиниран браузърен и документен обектни модели(частична диаграма) (Куигли, 2004, стр. 227)

По-долу ще бъдат описани малко по-подробно основните обекти, използвани при създаването на настоящата дипломна разработка.

#### **4.3.1.1.Обектът navigator**

Обектът navigator съдържа свойства и методи, които описват браузъра. Той може да бъде използван за проверка, специфична за различните платформи, за да се определи версията на браузъра, който се използва, какви plug-in-модули са налични и т.н.

Свойствата на обекта navigator са:

- `appName` – кодово име на браузъра;
- `appName` – име на браузъра:  
`ns=navigator.appName=="Netscape");`
- `appName` – версия на браузъра:  
`ie=(navigator.appName.toLowerCase().indexOf("microsoft")>-1);`
- `mimeType` – масив от MIME типове (Multi-purpose Internet mail extensions), поддържани от браузъра;
- `platform` – операционната система, в която се намира браузърът;

Браузърите поддържат различни характеристики, свойства и методи. С помощта на обекта navigator се открива информация за потребителския браузър, което позволява страниците да се генерират по начин, по който да бъдат възпроизведени най-добре от потребителския браузър.

Тъй като различните браузъри предлагат различни свойства и методи, манипулират по различен начин събитията, предлагат различни начини за позициониране на обектите и т.н., в дадена страница е наложително да се извършва проверка за вида и версията на браузъра, в който се разглежда дадената страница. Една от основните цели на тази дипломна работа е разработеният сайт да може да се отваря правилно в колкото се може повече различни браузъри и версии на браузъри. По тази причина изграждането на падащото меню в сайта е реализирано в няколко различни файла, всеки от които съдържа кода необходим за зареждане на менюто в съответния браузър.



В зависимост от това кой е текущият браузър и съответно коя негова версия се използва в началото на всяка една от страниците се идентифицира браузъра и се зарежда файл, в който са описани необходимите функции за изграждането на падащото меню:

```
....  
ns=navigator.appName=="Netscape";  
ns4=(ns &&(navigator.appVersion.indexOf("4.")!=-1));  
mac=(navigator.appVersion.indexOf("Mac")!=-1);  
ie=(navigator.appName.toLowerCase().indexOf("microsoft")>-1);  
ie5=(navigator.appVersion.toLowerCase().indexOf("msie 5.")!=-1);  
ie6=(navigator.appVersion.toLowerCase().indexOf("msie 6.")!=-1);  
ie4=((ie)&&!ie5)&&!ie6);  
....  
if((ns5)||((ie5)&&(mac)))  
    brn="ns6";  
document.write("<script          language=\"JavaScript1.2\"  
src=\""+DQM_codebase+"dqm_"+brn+".js\"></script>");.....
```

Основна разлика в браузърите е начина, по който се реализира позиционирането на елементите в документа. Например в Netscape (версия 4.0) това става с помощта на тага <layer> - прототип за CSS позициониране, за управление на позицията и видимостта на елементите, докато в Internet Explorer това е реализирано посредством all DOM (Document object model).

#### **4.3.1.2.Обектът window**

Обектът window е основният обект в йерархията от обекти в JavaScript и автоматично се дефинира за всеки отворен прозорец. Той има много свойства, които от своя страна също са обекти. Някои от тези свойства са многократно използвани в настоящият сайт.

- **closed** – връща истина, ако прозорецът е затворен;

- **defaultStatus** – подразбиращо се съобщение за състояние, визуализирано в лентата на състоянията в дъното на прозореца;
- **document** – всеки прозорец съдържа обект document, който съответства на HTML документа, показан в прозореца;
- **frames** – масив от фреймови обекти е рамките на прозореца;
- **history** – това е обект, съдържащ последно заредените URL адреси в прозореца. Той е най-използваният обект в JavaScript за придвижване напред и назад към страница, подобно на бутоните back и forward на брауъра. Той може да прави обръщение само към онези страници, които са били посетени – това са страниците, записани в неговия стек. Обектът има свойство length и методи go(), back() и forward(). Този обект е използван в разработката на сайта за придвижване между отделните страници:

... <a href="javascript:history.go(-1);" class="nav1">назад</a> ... – това е кодът на бутон „назад”, който се появява във всички страници от сайта.

```
... document.open();
```

```
document.write('<a href="#" onClick="history.back();return false;">'+
```

```
''+</a>');
```

```
document.close();....- този код се изпълнява при извикване на събитието onClick върху дадена снимка от страниците с картинната галерия за даден празник;
```

- **length** – броят на фреймовете в прозореца;
- **location** – използва се за осъществяване на достъп до URL-адреса на текущо заредения документ в прозорец. Тъй като той също е и отделен обект има свойства и метод. Този обект и неговите свойства са използвани многократно в разработването

на текущата дипломна работа, при създаването на падащото меню в сайта:

```
...if((where.length<1)||((where.toLowerCase()=="_self"))
    {
        window.location.href=eval("DQM_url"+index);
    }....
```

Това е част от функцията, с помощта, на която определяме кой файл трябва да покажем;

- **name** – името на прозореца;
- **offscreenBuffering** – управлява актуализирането на екрана;
- **opener** – прозорецът, който е отворил текущия прозорец;
- **parent** – показва прозорец, който съдържа друг прозорец;
- **screen** – визуализира информация за екрана;
- **self** - обръща се към текущият прозорец;
- **status** – задава временно съобщение в лентата за състоянията, в резултат от взаимодействие с потребителя;
- **top** – най-горният прозорец, съдържащ конкретен прозорец. Това свойство на обект сме използвали във всички страници на сайта, при натискането на бутона „нагоре” , за да се върнем в началото на текущата страница:  
...<a href="#top" class="nav1">нагоре</a>...;
- **window** – идентифицира текущия прозорец, към който се обръщаме;

#### **4.3.1.3.Обектът document**

Този обект се дефинира, когато в страницата се срещне HTML тага <body> и остава валиден, докато страницата в браузъра не се промени. Това е най-често използваният обект при изграждането на настоящия сайт.

Някои от свойствата на обекта document са:

- bgColor, fgColor – определя цвета на фона и текста, спрямо HTML тага <body>;
- cookie – позволява четене и запис на HTTP „бисквитки“;
- forms[] – масив от обекти forms, отнасящи се за HTML тага <forms>;
- images[] - масив от обекти images, отнасящи се за HTML тага <img>;
- linkColor, alinkColor, vlinkColor – определя цвета на непосетените, активните и посетените връзки, свързани с атрибутите на тага <body>;
- links[]- масив от обекти links, отнасящи се за HTML тага <a href>;
- title – заглавието на текущия документ;
- URL – Низ, съдържащ URL-адреса на документа и т.н.

Методи на обекта document, които са използвани често при създаването на сайта са:

- clear() – изчиства текущия документен прозорец;
- close() – затваря документния прозорец за писане;
- focus() – поставя фокуса върху документа;
- open() – започва нов документ, като изтрива стария;
- write() – записва и добавя текст към текущия документ;
- writeln() – същото като write(), но добавя нов ред, ако е в таг <pre>;

```
...document.write("<script                                language=\"JavaScript1.2\"  
src=\"\"+DQM_codebase+"dqm_"+"brn+".js\"></script>");  
document.close();....
```

#### 4.3.1.4.Обектът images

Обектът image е свойство на document, което предлага възможност за достъп до заредените в документа изображения.

Свойствата на обекта image съответстват на атрибутите на тага <img>:

- border – определя ширината на рамката на изображението;
- complete – определя дали изображението е свалено;
- height – определя височината на изображението;
- hspace – хоризонтално пространство около изображението;
- lowsrc – определя изображение по избор за визуализиране при устройства с ниска разделителна способност;
- name – име на изображението;
- src – низ, съдържащ пътят и името на изображението;
- vspace - вертикално пространство около изображението;
- width – ширина на изображението;

За предварително зареждане на изображения извън екрана, JavaScript осигурява конструктор за изображения. Той е особено полезен когато трябва да се заредят големи графични файлове, за свалянето на които е необходимо много време, или за изображения, които се сменят динамично в сайта. Израженията се зареждат предварително и са на разположение в кеш паметта, когато потребителят ги поиска – така времето за отговор значително намалява. Тази особеност на обекта Image е използвана в настоящата дипломна работа за разработката на падащото меню, видимо във всяка една от страници в сайта, тъй като отделните части на менюто са динамично сменящи се изображения:

```
...img01 = new Image();  
img01.src = "../images/1b.gif";  
img012 = new Image();
```

```
img012.src = "../images/1b_hl.gif";.....
```

Променяйки свойството src на дадено изображение, е възможно то динамично да бъде заменено от друго.

- **Обработка на събития в JavaScript (Манипулатори на събитията)**

Събитията в JavaScript са действия, предизвикани от потребителя – натискане на бутон, преместване на мишката върху изображение и т.н. Те са асинхронни, т.е могат да настъпят по всяко време. При настъпване на събитие JavaScript може да реагира като изпълни действия, предизвикани от събитието, които ще внесат промяна в страницата.

Манипулаторите на събитията в JavaScript се разполагат между таговете <script> и </script>. Събитията се асоциират с HTML таговете и се разполагат между съответния таг – ако събитието е предизвикано от натискане с мишката върху връзка то ще бъде атрибут на тага <a href...> и т.н. Низът, задаващ манипулатора на събитие е команда, която трябва да се изпълни при задействането на това събитие от потребителя . Тази команда може да бъде както функция, вградена в JavaScript, така и функция, дефинирана от потребителя.

Докато свойствата и методите могат да бъдат асоциирани само с един обект, събитията обикновено се асоциират с колекция от обекти.

Най-често използваните манипулатори на събития са:

- onAbort – събитие, което се отнася за изображения и настъпва, когато дадено изображение се прекъсне;
- onChange – използва се с полета за въвеждане на информация от потребителя, полета за избор. Настъпва когато потребителя промени стойност на елемент и премести фокуса върху друг. Използва се предимно за проверка на валидност на въведените данни;

- `onClick` – отнася се за връзки, бутони, карти на изображения и т.н. Събитието настъпва , когато потребителят щракне върху обект. Върнат стойност `false` ще отмени подразбиращото се действие;
- `onMouseOver` – асоциира се с връзки и изображения с връзки. Настъпва при преместване на мишката върху връзката. Връща `true`, за да предотврати показването на връзката в лентата на състоянието;
- `onMouseOut` – подобно на `onMouseOver`, но настъпва при преместване на мишката извън очертанията на връзката/изображението ;
- `onSubmit` – настъпва при изпращане на формуляр на сървър. Връща `false` за предотвратяване на предаването към сървър;
- `onLoad` – настъпва след зареждане на изображение или документ;
- `onResize` – настъпва при промяна на размера на прозореца на брауъра и т.н.;

В създаването на манипулатор на събитие съществуват два етапа:

- Манипулаторът се задава както атрибут на HTML тага на документ, връзка или формуляр;
- Присвояване на стойност на манипулатора на събитието – вградена или потребителска функция. Ако функцията е потребителска тя трябва да е дефинирана или в програма на JavaScript или като директни скриптови конструкции.

Манипулаторите на събития `onMouseOut` и `onMouseOver` са често използвани за създаване на ролувъри - изображение, което се заменя от друго изображение всеки път, когато курсорът на мишката попадне върху връзка или изображение. В настоящата дипломна работа описаните манипулатори са използвани с тази цел. Всеки път, когато потребителят попадне в рамките на някое от изображенията на основното меню, то се сменя с друго изображение – надписът е същият, но цветовете използвани като фон и цвета на текста са

различни. По този начин потребителят се ориентира много по-лесно къде точно се намира показалеца на мишката в даден момент, за да може да избере желаното от него подменю, чийто детайли иска да разгледа.

В настоящата дипломна работа роловер изображенията са реализирани по следния начин:

```
... <td><a href="sub_6.php" ONMOUSEOVER="OnImage('img06');"
ONMOUSEOUT="OffImage('img06');"></a></td>...
```

Когато показалеца на мишката се намира върху даденото изображение се изпълнява функцията `OnImage(name)`, дефинирана в `<head>` блока на документа, чиято цел е да намери и покаже новото изображение в страницата:

```
function OnImage(name) {
    if (document.images) {
        fullname = eval(name + "2");
        document[name].src = fullname.src;
    }
}...
```

По подобен начин е реализирана функцията `OffImage(name)`:

```
function OffImage(name) {
    if (document.images) {
        fullname = eval(name);
        if (fullname.complete) {
            document[name].src = fullname.src;
        }
    }
}...
```

С помощта на конструктора на изображения, изображенията използвани в падащото меню са заредени предварително:

```
...img06 = new Image();
img06.src = "../images/6b.gif";
img062 = new Image();
```



```
img062.src = "../images/6b_hl.gif";...
```

Гореописаните манипулатори на събитията са използвани за “статичната” част от менюто, т.е това са описанията на “плаващите” празниците, чиято дата не е точно фиксира в календара. Тези празници нямат техни подчини и не е необходимо да се разтваря падащо подменю при преминаването с курсора на мишката върху тях.

Падащите менюта в настоящата дипломна работа също са реализирани с помощта на манипулаторите на събитията в JavaScript:

```
... <td></td>...
```

Когато потребителят позиционира курсора на мишката върху някое от изображенията, например върху надписа “зимни”, под него се отваря падащо подменю, което показва подкатегориите, от които потребителят може да избира. Изображенията за различните категории също се променят при преминава с мишката върху тях, като по този начин улеснява потребителя. При преместването на мишката извън очертанията на дадено изображение съответното подменю се скрива, то вече не е необходимо, тъй като потребителят е насочил вниманието си към някоя друга част от сайта.

Свиването и разгъването на падащите менюта са реализирани с функциите `hideMenu(event)` и `showMenu(event)`, които са описани във външен файл, включен във всеки от документите, в които е нужен.

Тъй като всеки HTML до голяма степен е набор от елементи, наречени тагове, с помощта на DOM(Документен обектен модел) всеки един от тези елементи е представен като възел в йерархичното дърво, предложено от този модел. Някои от най-важните свойства на HTML елементите, които са използвани в настоящата разработка са:

- `id` - атрибутът `id` се използва за идентифициране на всеки елемент като уникален блок, така че да може да бъде манипулиран от функция на JavaScript;

- style - това е стойността на вграден (inline) атрибут за стил на CSS, чрез който могат да бъдат зададени размери и позиция на елементите, да бъдат дефинирани като видими или невидими.

```
... "style='position:absolute;left:"+bw+";top:"+bw+";...
```

```
... "<img id=qmim"+j+" style='position:absolute;cursor:hand;z-  
index:9;visibility:hidden;' src='"+eval("DQM_rollover_image"+j)+"  
width="+imgwh[0]+" height="+imgwh[1]+" border=0'  
onclick=\"q32(\""+j+"\")\">";.....
```

Един от най-често използваните методи при манипулирането на HTML елементите е getElementById(). Той връща обръщение към HTML елемента, който след това може да бъде обработен с JavaScript функция.

```
....return eval("document.getElementById(\""+id+"\")");...
```

Манипулаторите в JavaScript са използвани и при създаването на файловете-шаблони, които определят външния вид на форума, достъпен от настоящия сайт. При регистрирането на потребител във форума, той трябва да избере държавата, в която се намира . При избора на дадена държава от скролиращ списък с държави името на тази държава се изписва в текстово поле:

```
... <td class=caption5><input type=text name=from maxlength=100 size=20  
value="{ $from}" class=textForm>  
<select name=fromSelect class=textForm  
onChange="JavaScript:insertCountry();" style="width:200px;">...
```

Когато настъпи събитието onChange се извиква и изпълнява JavaScript функцията insertCountry(), дефинирана в <head> частта на файла **user\_dateform.html**:

```
function insertCountry() {  
    var ind;  
    ind = document.regform.fromSelect.selectedIndex;  
    ind = document.regform.fromSelect.options[ind].value;  
    document.regform.from.value = ind;}  
}
```

При изпълнението на тази функция се извлича индекса на избраната от списъка държава, извлича се съответната стойност от списъка и тя се присвоява на текстовото поле form.

#### **4.4.PHP**

PHP (PHP Hypertext Preprocessor) е вграден език за създаване на скриптове от страна на сървъра. Това означава, че той работи в рамките на HTML документ, добавяйки към него възможността за генериране на съдържание по желание на потребителя. С негова помощ можем да конвертираме един сайт в web-приложение, което не е просто съвкупност от статични страници с информация, и което внася динамичност в сайта.

Някои от основните предимства на PHP са:

- простота;
- почти естествен начин за използване на бази данни и независимост от платформата;
- той е с отворен код – характеристика, която не е за пренебрегване;

В настоящият сайт PHP е използван основно при разработването на форума, прикрепен към дадения сайт и ще бъде подробно описан в дипломната работа на Надежда Каменова Иванова. Тук ще отбележим само начина, по който PHP е използван в информационните страници на сайта. Тъй като всички те съдържат общи части при създаването на файловете се възползвахме от възможността на PHP за повторно използване на код. PHP улеснява използването на части от код или HTML в един документ с помощта на конструкциите include и require. Разликата между тях е особено важна. Конструкцията include е условна – файлът се чете при всяко срещане на include, т.е ако конструкцията не бъде достигната, файлът няма да бъде включен. Но ако един и същ файл е включен два пъти с конструкцията include, PHP ще го прочете два пъти. (Кастането, 2001, стр.775) Включените с помощта на require файлове се прочитат само веднъж. По този начин се

улеснява и бъдещото преправяне на страниците в сайта, тъй като това ще трябва да се направи на едно място, а не във всяка една от страниците, които са много на брой.

## **5.Използвани инструментални средства**

### **5.1.Adobe PhotoShop**

Adobe PhotoShop е сред най-известните програми за обработка на растерни изображения. PhotoShop е предназначен за обработка и създаване на снимки, скици, колажи и тн. С тази програма могат да се редактират изображения, да се променят фотографии и други сканирани материали, да се добавят специални ефекти, да се прехвърля едно изображение към друго, да се помества текст, да се нагласят цветовете. Той също така осигурява възможност за вмъкване на изображение от скенер – функция, която многократно е използвана в създаването на състоящият сайт. Сканираните снимки са обработвани и след това експортирани в **JPEG** формат.

Някои от функциите, предлагани от PhotoShop са:

- изрязване на част от графики и картини – функция, която е използвана за подбиране на точно определена част от сканираните снимки;
- използване на слоеве (layers);
- прилагане на различни ефекти върху обектите – илюзия, прозрачност, която е една от силните страни на Adobe Photoshop;

Голяма част от сканираните снимки бяха подложени на допълнителна обработка и вмъкване на ефекти.

- обработка и експортиране в различни файлови формати е голямо предимство в работата с Adobe Photoshop.

Файловете формати са изключително важни при подготовката на web документа. Основните графични файлови формати използващи се в html документи са:

- **GIF** (Graphics Interchange Format ) - доминиращ формат за изобразяване на графиките в Web страниците. Това се дължи на възможността на GIF формата да компресира графики до 256 цвята без загуби, със степен на компресия 4:1. Изходните файлове са малки и лесно преносими. GIF изображенията опростяват визуалните ефекти, имат прозрачни фонове, които позволяват лесното им вписване в друга структура.

- **JPEG** (Join Photographic Experts Group) стандартът поддържа 24 – битов цвят. Форматът изпуска информация за цветовете, дължащо се на използването на метод за компресиране със загуби. Важна черта е, че степента на загуба с тази компресия може да бъде променяна, като се настройват параметрите за компресия. Препоръчва се за повечето висококачествени графични продукти. Той е със сравнително по-добро качество на компресия от GIF формата и същевременно физическия обем на файла не е голям, което го прави подходящ за HTML езика.

Някои от останалите формати, които се поддържат от Adobe Photoshop са:

- PSD, PDD, EPS – формати на Photoshop;
- IFF, BMP, PIC, PNG и т.н. – формати с високо качество на изображението, но които не са подходящи за HTML поради значителният обем на файла. Използват се предимно за печатна дейност.

На базата на тази програма бяха създадени и обработени голяма част от снимките в сайта.

## **6.Тестване, оценка и усъвършенстване на сайта**

За разработване на сайта за българските традиции е необходимо изучаване на поведението и начина на работа на различните браузъри, както и познания в областта на различните технологии като Hypertext Markup Language (HTML), скриптове, Cascading Style Sheets (CSS), web-сървъри, Базии данни и PHP.

Тестването на сайта бе направено на следните браузъри – Internet Explorer 5.0, Internet Explorer 5.5, Internet Explorer 6.0, Netscape Navigator 4.7, Netscape Navigator 6.0, Opera 7.03 и Mozilla FireFox 1.0 . Тъй като CSS не се поддържа напълно от всички браузъри някой от дефинираните стилове не се визуализират коректно. Примерен проблем е визуализирането на дефинициите за цвят на скрол бара. Те се показват коректно единствено в Internet Explorer с версии по – високи от 5.0. Тествахме сайта на различни резолюции и монитори. Тъй като съдържанието е заключено в HTML таблици с фиксирана ширина не открихме проблем. Евантуално некоректно визуализиране на страниците би се получило и при минимизиране и максимизиране на прозореца на дадения браузър, ако съдържанието не е поставено в таблици. За коректното появяване на уводната анимация съответния браузър трябва да има инсталиран Flash Player. При тестването на различни компютри се установи, че при различните настройки на браузърите се получаваше разместване на шрифтовете и полетата във формите спрямо изображенията. Този проблем се реши след като фиксирахме големината на шрифтовете в пиксели.

Регистрацията и влизането на даден потребител с неговото име и парола във страниците на форума не функционира, ако на потребителската машина е инсталирана защитна стена ( fire wall ), която забранява сесиите. Голяма част отделихме за тестване дали данните въведени от потребителите са коректни. Например дали въведен валиден е – mail адрес. Версията на PHP трябва да е 4.0 или по – късна и да поддържа сесии.

Тестовите на разработката са извършени на Windows 98, Windows 2000, Windows XP, Apache/1.3.31, PHP/4.3.9, MySQL 3.29.49.

Оценката на сайта според нас е по – скоро положителна. Едни от главните предимства на сайта са :

- неговия дизайн – умелото подбиране на цветовата гама и графичните ресурси.
- лесна и интуитивна навигация.
- пълнота на информацията.
- бърза скорост на зареждане, благодарение на оптимизирането на снимковия материал.

Както всяка една разработка сайтът има и недостатъци. След направеното тестване и оценка разграничихме следните насоки, в които може да се работи за усъвършенстване на сайта:

- превод на наличната информация на други от широко разпространените езици – немски, френски, испански.
- могат да се добавят препратки към други сайтове с подобно съдържание
- би могло да се добави търсене по име, което да намира деня (празника), в който хората с това име празнуват именния си ден.
- може да се добави още снимков материал.
- смяната на езика би могла да е достъпна от всяка страница на сайта, не само от началната.

Сайтът за българските традиции и обичаи е достъпен от интернет адресът <http://traditions-bg.com> .

## **7.3. Заключение**

Разработеният сайт за българските традиции и обичаи е една малка част от по-голям проект, обхващащ всички аспекти от наследството на България. Основната цел на тази дипломна работа е да запознае читателя с оцелялото народно богатство на българския народ. Описани са голям брой от обичаите и традициите на българите от началото на миналия век - църковните празници, както и езическите обреди, запазили се и до днешни дни. Сайтът съдържа подробна информация на български и английски език, която се базира на голям брой изследвана книжна литература, както и информация, предложена в електронен вид.

За реализирането на настоящият сайт са използвани следните технически средства - HTML, CSS, Java Script, Flash, PHP и MySQL.

Особено внимание е обърнато на дизайна на сайта - избора на цветове, представяне на графичната информация и т.н, с които сайтът превъзхожда намеренията в интернет пространството сайтове на подобна тема.

В началото на дипломната работа бяха поставени няколко цели и изисквания, на които се стремим да отговаря сайта и които смятаме, че успяхме за да изпълним с настоящата разработка. В сравнение с останалите сайтове по аналогична тематика в българското и чуждестранното Интернет пространство може да се каже, че изготвеният сайт се нарежда на едно от първите места по атрактивност, пълнота и лесна навигация. Естествено сайтът има и недостатъци, които както и предимствата са посочени в предишната точка.

Въз основа на направения анализ и представеният проект се надяваме уважаемата комисия и рецензент да оценят труда ни и този на научната ни ръководителка по достойнство.

Искаме още веднъж да благодарим най-сърдечно на научната ни ръководителка Евгения Ковачева, на която държим изключително много.



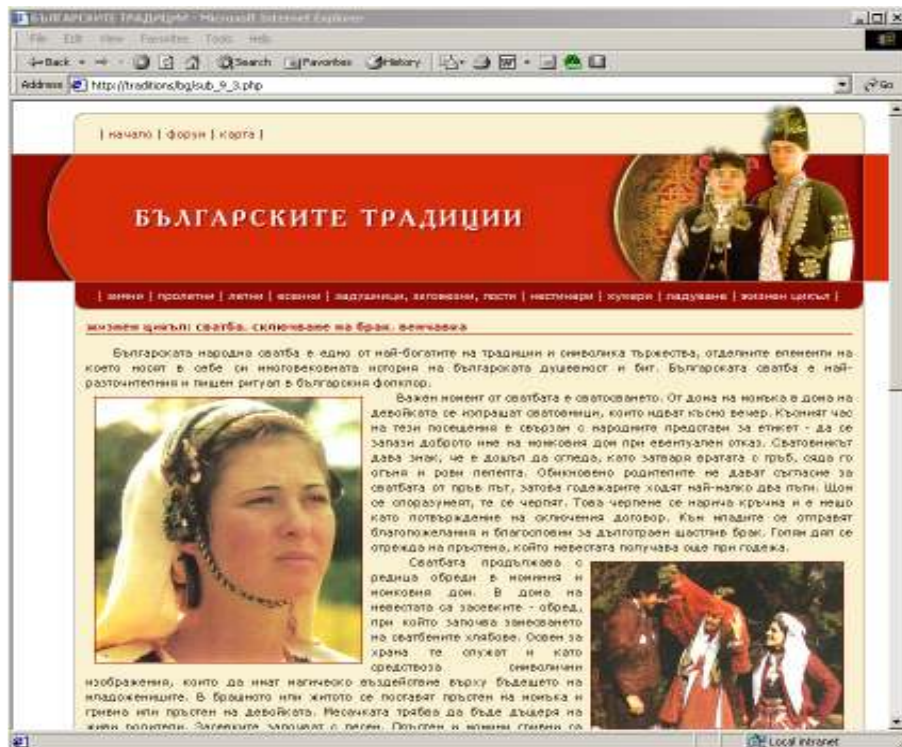
## **8.Използвана литература**

1. книга: Ели Куигли (2004), JavaScript в примери, СофтПрес; София.
2. книга: Хесус Кастането,Хариш Роат, Саша Шуман, Крис Сколо, Дийпак Вилайт (2001), Професионално програмиране с PHP, СофтПрес; София.
3. интернет адрес: <http://freewarejava.com/javascript/index.shtml>

**9. Приложение**



*1. Начална страница на сайта*



*2. Вътрешна страница на сайта*



*3. Картинна галерия на сайта 1*

## WEB БАЗИРАНО ПРЕДСТАВЯНЕ НА ТРАДИЦИИТЕ И ОБИЧАИТЕ НА БЪЛГАРИЯ

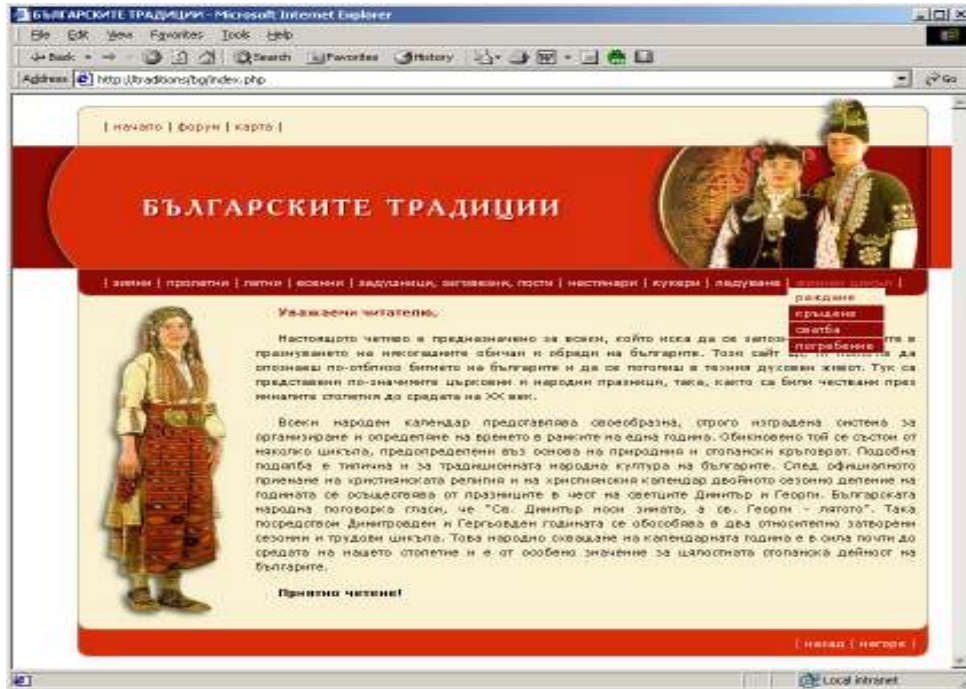


4. Картинна галерия на сайта 2

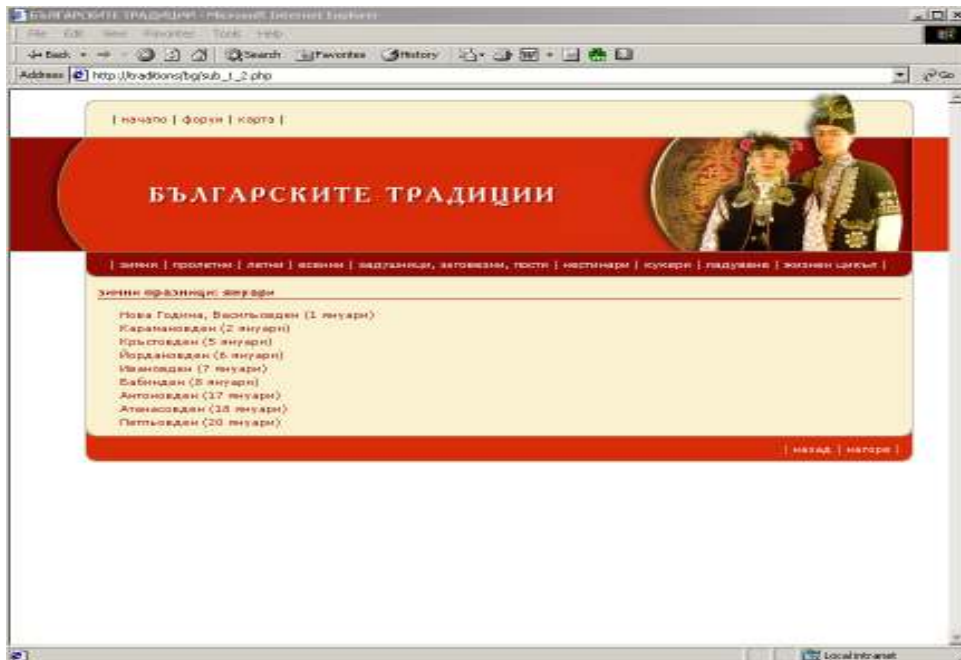


5. Карта на сайта

## WEB БАЗИРАНО ПРЕДСТАВЯНЕ НА ТРАДИЦИИТЕ И ОБИЧАИТЕ НА БЪЛГАРИЯ



6. Разгъване на менюто



7. Под линкове на раздел от менюто