



**Софийски университет „Св. Кл. Охридски“**

**Факултет по математика и информатика**

*Специалност*

*„Софтуерно инженерство“*



## **Курсов проект**

**XML технологии за семантичен уеб**

*Зимен семестър, 2023/2024 год.*

**Тема № 48:**

**„Каталог чудесата на света - 2“**

**Автори:**

Христина Гаджева, Ф№ 7MI0600131

Николай Николаев, Ф№ 8MI0600010

**Ръководители:**

доц. д-р Александър Димов

гл. ас. д-р Явор Данков

януари, 2024

София

## Съдържание

### Съдържание

|      |                                                      |    |
|------|------------------------------------------------------|----|
| 1.   | Въведение.....                                       | 3  |
| 1.1. | Актуалност и проблем.....                            | 3  |
| 1.2. | Използвани технологии .....                          | 3  |
| 1.3. | Структура на документа.....                          | 3  |
| 2.   | Анализ на решението .....                            | 4  |
| 2.1. | Работен процес.....                                  | 4  |
| 2.2. | Структура на съдържанието .....                      | 5  |
| 2.3. | Тип и представяне на съдържанието .....              | 9  |
| 3.   | Дизайн.....                                          | 10 |
| 3.1. | Изпълнение на условията.....                         | 10 |
| 3.2. | Архитектура на решението и обяснения по кода.....    | 17 |
| 4.   | Тестване .....                                       | 28 |
| 4.1. | Microsoft Edge и Google Chrome.....                  | 28 |
| 4.2. | Тестване с IE.....                                   | 31 |
| 4.3. | Тестване с Mozilla Firefox .....                     | 33 |
| 5.   | Заклучение и възможно бъдещо развитие .....          | 34 |
| 5.1. | Възможности за бъдещо развитие .....                 | 34 |
| 6.   | Разпределение на работата .....                      | 35 |
| 7.   | Използвани литературни източници и Уеб сайтове ..... | 35 |
| 8.   | Апендикс.....                                        | 37 |

## 1. Въведение

Текущият документ представлява документация към изготвения курсов проект по XML технологии за семантичен уеб на тема 48 „Каталог на чудесата на света - 2“.

### 1.1. Актуалност и проблем

Проектът представя Чудесата на света, разпределени по три категории – Старите 7 чудеса на света, Новите 7 чудеса на света и седемте най-известни природни забележителности. За всеки един обект е подробно описана историята, географско положение, начин на пътуване, входни такси и други интересни факти за чудесата. Целта на проекта е да представи необходимата информация на едно място и да улесни бъдещите туристи като спестява нуждата от търсенето по различни сайтове и загубата на време. Също така има задачата да разпространи интересни и любопитни факти за всяко едно чудо, за да привлече вниманието на хората и да ги насърчи да посетят конкретното място. За всеки един обект са предложени и други забележителности в околност, което ще допринесе за по-пълноценна и добра екскурзия към вече избраната дестинация.

### 1.2. Използвани технологии

Технологиите и езиците са базирани на XML. Това дава възможност за добре структурирани и валидирани данни чрез XML документ, вътрешно DTD и XML Schema. За представянето на информацията в браузър се използват езиците HTML и CSS, а чрез XSLT се трансформират данните от XML формат в HTML. Обектите, описани вътре са сортирани по вид и по местоположение, за да може потребителите да се ориентират по лесно.

### 1.3. Структура на документа

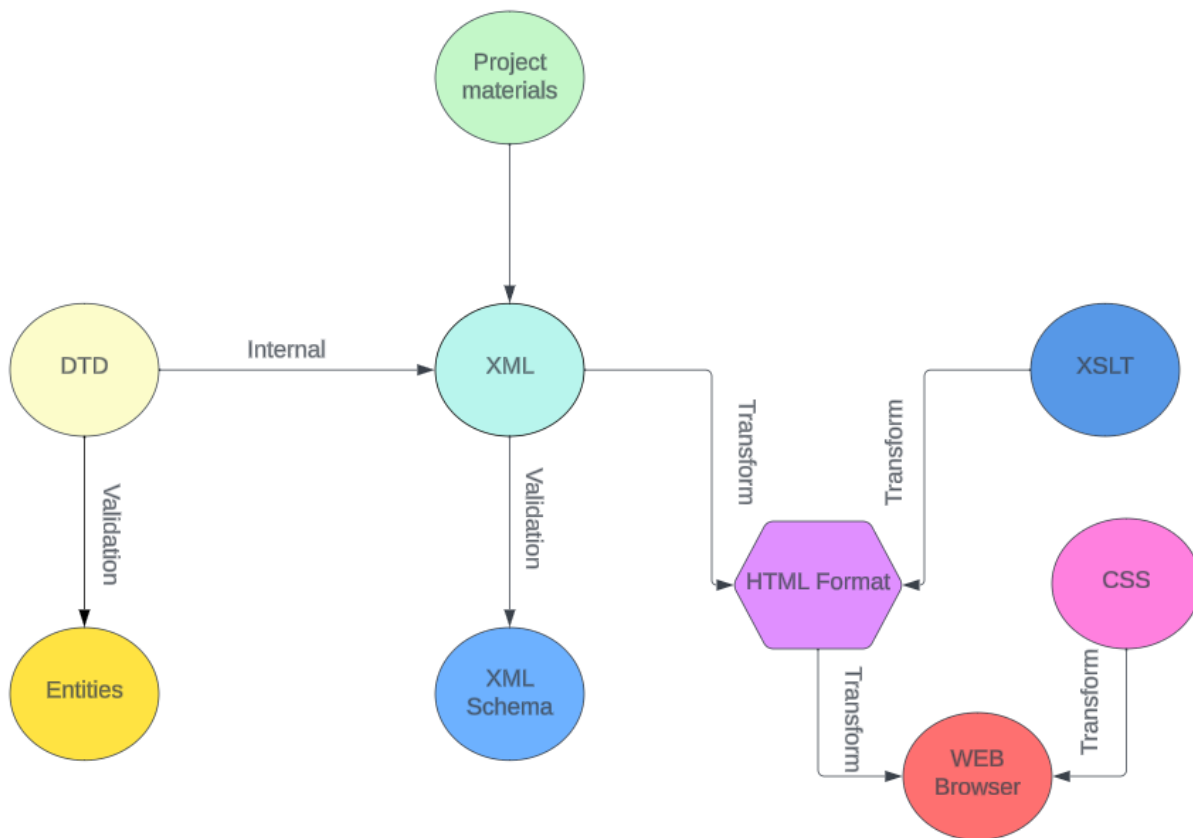
Структурата на документацията представя информация за процеса по разработването на курсовото задание. Оформянето ѝ се базира на следните стъпки:

- Въведение - тук е упомената основната тема на проекта, проблемът, свързан с нея, използвани технологии за решението му.
- Анализ на решението - съдържа осъществяването на работния процес, структурата на самия проект типа на използваното съдържанието.
- Дизайн - показва как се използват XML технологиите, визуализира използвания код и описва приложението му.
- Тестване - представя информация за вида документи и браузъри, на които е тестван проекта, и изхода от крайния резултат.
- Заключение и възможно бъдещо развитие - съдържа предимствата и недостатъците на проекта и се дават насоки за просперитета на проекта в бъдеще.
- Разпределение на работа - посочено е разпределението на работния процес между участниците в екипа.
- Използвани източници - посочени са използваните източници за изготвянето на курсовата работа.
- Апендикс - съдържа допълнителна информация.

## 2. Анализ на решението

### 2.1. Работен процес

Съдържанието на каталога е представено под формата на XML документ, като графичните елементи са добавени с вътрешно DTD чрез entities. За информацията в сайта са използвани надеждни интернет източници и официални сайтове на обектите, които предоставят актуална информация за работно време, цена и т.н. Снимките също са взети от Интернет, а след това са селектирани в папки спрямо категориите - природни чудеса, нови чудеса и стари чудеса. Валидирането на структурата и съдържанието на XML документа е извършено чрез подходящо изградена за целта XML схема. Представянето на XML съдържанието под формата на страница в уеб браузър е направено с помощта на XSL и CSS документи, описващи трансформацията на информацията и средата на представяне. Чудесата на света са селектирани по различни категории - по вид, по континенти и по азбучен ред. Работният процес преминава през следните етапи - извличане, събиране, селектиране и обработка на текстов и снимков материал, въвеждане на данните в XML документ, съобразно синтактичните норми и правила на езика, валидацията на документа чрез написана XML Schema и вътрешно DTD, трансформация на информацията в HTML формат с напавата на XSL файл и създаването на CSS документ, служещ за визуализацията на проекта на уеб браузър.



## 2.2. Структура на съдържанието

0. catalog – коренов елемент на XML документа. Той съдържа елементите: continents, types\_of\_wonders, wonders. Първите два поделементи служат за селектиране на чудесата по континенти и по вид на чудото, а wonders е основен елемент, който съдържа всичките 21 чудеса.

1. continents – пряк наследник на catalog елемента. Негов подемент е continent\_selector с атрибути continentID и wonderRef1. Целта на тази секция е да групира чудесата на света по местоположението им според 7-те континента.

1.1. continent\_selector - пряк наследник на continents елемента. Негов подемент е continent\_name, който носи информация за името на съответния континент. Елементът continent\_selector има два атрибута: continentID и wonderRef1.

1.1.1.continentID – атрибут на елемента continent\_selector. Този атрибут е от тип ID и е означен като задължителен. Всеки continent\_selector има уникално continentID, което го идентифицира и улеснява връзката с xsl документа.

1.1.2.wonderRef1 – атрибут на елемента continent\_selector. Този атрибут е от тип IDREFS и е означен като опционален. Това означава, че съдържа много референции, всяка от които се свързва с уникалния wonderID на съответното чудо. По този начин се изгражда връзка между континентите и чудесата, които се намират в него.

1.1.3.continent\_name - пряк наследник на continent\_selector елемента. Елементът continent\_name е от тип PCDATA. Той съдържа името на съответния континент.

2. types\_of\_wonders – пряк наследник на catalog елемента. Негов подемент е type\_of\_wonder с атрибути type\_of\_wonder\_ID и wonderRef2. Целта на тази секция е да групира чудесата на света по видове - природни, нови и стари.

2.1. type\_of\_wonder - пряк наследник на types\_of\_wonders елемента. Негов подемент е type\_of\_wonder\_name, който носи информация за името на съответния континент. Елементът type\_of\_wonder има два атрибута: type\_of\_wonder\_ID и wonderRef2.

1.1.1.type\_of\_wonder\_ID – атрибут на елемента type\_of\_wonder. Този атрибут е от тип ID и е означен като задължителен. Всеки type\_of\_wonder има уникално type\_of\_wonder\_ID, което го идентифицира и улеснява връзката с xsl документа.

1.1.2.wonderRef2 – атрибут на елемента type\_of\_wonder.. Този атрибут е от тип IDREFS и е означен като опционален. Това означава, че съдържа много референции, всяка от които се свързва с уникалния wonderID на съответното чудо. По този начин се изгражда връзка между континентите и чудесата, които се намират в него.

1.1.3. `type_of_wonder_name` - пряк наследник на `type_of_wonder` елемента. Елементът `type_of_wonder_name` е от тип PCDATA. Той съдържа името на съответния вид чудеса.

1. `wonder` - пряк наследник на `catalog`. Има атрибут `wonderID` и поделементи `name`, `destination`, `history`, `description`, `accessibility`, `related_marks`, `images`. Съдържа подробна информация за всяко от чудесата
  - 1.1. `wonderID` - уникален идентификатор за всяко чудо. Служи за осъществяването на връзката между чудото и категорията, към която принадлежи. Участва в проверката за уникалност на чудото.
  - 1.2. `name` - поделемент на `wonder`. Тук е записано името на чудото. Участва в проверката за уникалност на чудото.
  - 1.3. `destination` - поделемент на `wonder`. Съдържа поделементи - `continent`, `country`, `region`, `state`, `city`, `place`, `cont_id`, `type_id`. Дава точна информация за местоположението на обекта. Не е задължително всеки поделемент да го има, някои може и да не присъстват на всички чудеса.
    - 1.3.1. `continent` - поделемент на `destination` - името на континента, където се намира чудото
    - 1.3.2. `country` - поделемент на `destination` - името на държавата, в която се намира забележителността
    - 1.3.3. `region` - поделемент на `destination` - региона, в който се намира обекта
    - 1.3.4. `state` - поделемент на `destination` - щатът, в който се намира чудото, ако е в САЩ
    - 1.3.5. `city` - поделемент на `destination` - градът, в който се намира забележителността
    - 1.3.6. `place` - поделемент на `destination` - мястото, на което се намира обекта
    - 1.3.7. `cont_id` – осъществява връзка между конкретното чудо и ключът (континентът), с който се свързва.
    - 1.3.8. `type_id` - осъществява връзка между конкретното чудо и ключът (видът на чудо), с който се свързва.
  - 1.4. `history` - поделемент на `wonder`. Има два поделемента - `creation` и `changes`. Предоставя информация за историята на обекта.
    - 1.4.1. `creation` – поделемент на `history`. Съдържа данни за създаването на забележителността - кога е създадена, къде и кой е изобретателят.
    - 1.4.2. `changes` - поделемент на `history`. Тук са разказани промените през годините, най-често срещани от природни бедствия или просто реставриране и модернизиране на забележителността.
  - 1.5. `description` - поделемент на `wonder`. Дава кратко описание за чудото.
  - 1.6. `accessibility` - поделемент на `wonder`. Съдържа три елемента - `transport`, `worktime` и `ticket_price`. Информира ни как да достъпим чудото и получаваме актуална информация за работно време и цени на обекта.
    - 1.6.1. `transport` - поделемент на `accessibility`. Показва видовете транспорт до мястото от близки точки и централни градове до него.

- 1.6.2. `worktime` - поделемент на `accessibility`. Дава актуален график на работното време.
- 1.6.3. `ticket_price` - поделемент на `accessibility`, съдържа информация за ценовите категории на забележителността.
- 1.7. `related_landmarks` - поделемент на `wonder`. За поделемент има `landmark` като може да има няколко забележителности, но може и да няма нито една.
  - 1.7.1. `landmark` - поделемент на `related_landmarks`, съдържа информация за други забележителности в околност на чудото, ако има такива
- 1.8. `images` - поделемент на `wonder`, който има поделемент `image`
  - 1.8.1. `image` - наследник на `images`. Съдържа поделемент `image_src`. Има атрибут `type`, защото в последствие може да се добавят още изображения и да имат различна номерация на типовете.
    - 1.8.1.1. `image_src` поделемент на `image` с атрибути `href` и `ent` като `href` дава пряк път до файла на изображение, а `ent` дава `entity` на изображението. `href` се използва при отварянето на сайта в FireFox, защото този браузър не поддържа функцията за `entities`, а `ent` референцията се използва за отварянето на проекта през google chrome и Edge.

За по-нагледно представяне снимков материал за структурата и сортировките на документа

```

<catalog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="wonders_of_the_world_edge.xsd">

    <wonders>
        <wonder wonderId=".....">
            <name>.....</name>
            <destination>
                <continent>.....</continent>
                <country>.....</country>
                <city>.....</city>
                <place>.....</place>
                <cont-id>.....</cont-id>
                <type-id>.....</type-id>
            </destination>
            <history>
                <creation>.....</creation>
                <changes>.....</changes>
            </history>
            <description>.....</description>
            <accessibility>
                <transport>.....</transport>
                <worktime>.....</worktime>
                <ticket_price>.....</ticket_price>
            </accessibility>
            <related_landmarks>
                <landmark>.....</landmark>
                <landmark>.....</landmark>
                <landmark>.....</landmark>
            </related_landmarks>
            <images>
                <image type="1">
                    <image_src href="....." ent="....."/>
                </image>
            </images>
        </wonder>
    </wonders>

</catalog>

```

Фиг. 1 Структурата на XML документа



```

<continents>
  <continent_selector continentID = "AS" wonderRef1="PETRA GREAT_WALL TAJ_MAHAL EVEREST HANGING_GARDENS_OF_BABYLON TEMPLE_OF_ARTEMIS_AT_EPHEBUS">
    <continent_name>Азия</continent_name>
  </continent_selector>

  <continent_selector continentID = "AU" wonderRef1="GREAT_BARRIER_REEF">
    <continent_name>Австралия</continent_name>
  </continent_selector>

  <continent_selector continentID = "EU" wonderRef1="COLLOSEUM STATUE_OF_ZEUS_AT_OLYMPIA COLOSSUS_OF_RHODES MAUSOLEUM_AT_HALICARNASSUS">
    <continent_name>Европа</continent_name>
  </continent_selector>

  <continent_selector continentID = "SA" wonderRef1="CHICHEN_ITZA STATUE_CHRIST MACHU_PICCHU RIO_DE_JANEIRO_HARBOUR">
    <continent_name>Южна Америка</continent_name>
  </continent_selector>

  <continent_selector continentID = "NA" wonderRef1="GRAND_CANYON PARICUTIN">
    <continent_name>Северна Америка</continent_name>
  </continent_selector>

  <continent_selector continentID = "AF" wonderRef1="VICTORIA_FALLS GREAT_PYRAMID_OF_GIZA LIGHTHOUSE_OF_ALEXANDRIA">
    <continent_name>Африка</continent_name>
  </continent_selector>

  <continent_selector continentID = "AN" wonderRef1="AURORA">
    <continent_name>Антарктида</continent_name>
  </continent_selector>
</continents>

<types_of_wonders>
  <type_of_wonder type_of_wonder_ID = "OW" wonderRef1="HANGING_GARDENS_OF_BABYLON STATUE_OF_ZEUS_AT_OLYMPIA TEMPLE_OF_ARTEMIS_AT_EPHEBUS MAUSOLEUM_AT_HALICARNASSUS
    LIGHTHOUSE_OF_ALEXANDRIA COLOSSUS_OF_RHODES GREAT_PYRAMID_OF_GIZA">
    <type_of_wonder_name>Стари чудеса на света</type_of_wonder_name>
  </type_of_wonder>

  <type_of_wonder type_of_wonder_ID = "NW" wonderRef2="STATUE_CHRIST GREAT_WALL PETRA CHICHEN_ITZA MACHU_PICCHU TAJ_MAHAL COLLOSEUM">
    <type_of_wonder_name>Нови чудеса на света</type_of_wonder_name>
  </type_of_wonder>

  <type_of_wonder type_of_wonder_ID = "NAW" wonderRef2="GRAND_CANYON AURORA EVEREST PARICUTIN RIO_DE_JANEIRO_HARBOUR VICTORIA_FALLS GREAT_BARRIER_REEF ">
    <type_of_wonder_name>Природни чудеса</type_of_wonder_name>
  </type_of_wonder>
</types_of_wonders>

```

Фиг. 2 Сортиране на съдържанието на XML документа по вид и по континент

## 2.3. Тип и представяне на съдържанието

За представянето на информацията за проекта са използвани 4 текстови файла и 21 графични изображения (.jpg).

2.3.1. Текстовите файлове са 4 на брой, като в тях са написани кодовете от xml документа, xml схемата, xslt файла и css частта от проекта.

2.3.1.1 wonders\_of\_the\_world.xml – в главната директория на проекта се намира XML документ, който е кодиран с UTF-8 и използва XML Version 1.0. Там е описано подробно цялостното съдържание на проекта, което следва добрите практики за форматиране, съобразно спецификите на езика. В началото на документа е включено вътрешно DTD - чрез него се представят графични изображения посредством XML entities. Документът декларира използване на XML schema, която

служи за валидация, и е направена необходимата връзка към XSLT файла за извличане на нужната информация.

2.3.1.2. `wonders_of_the_world.xsd` - разположен в главната директория на проекта, съдържа дефиниции за валидация на XML документа чрез използването на XML Schema. Този файл играе ролята на схема, която уточнява и задава структурата, типовете данни и ограниченията за елементите и атрибутите, които могат да бъдат използвани в основния документ. По този начин схемата предоставя стриктни правила и стандарти за валидация, които осигуряват съответствие с очакваната структура на данните в XML документа.

2.3.1.3 `wonders_of_the_world.xsl` - разположен в главната директория на документа. Чрез този XSLT файл се извършва трансформация на основния XML документ в HTML формат. Тук е описано преобразуването и форматирането чрез правилата за извличане, филтриране и форматиране на информация, благодарение на които основният файл е представен по желания начин в HTML изхода. Това предоставя възможност за представяне на проекта на различни уеб браузъри или на други средства за визуализация.

2.3.1.4 `wonders_of_the_world.css` - разположен в главната директория на документа. Има за цел да дефинира стиловете и външния вид на съдържанието, което бива представено в уеб браузъра, след трансформацията на XML чрез документа XSLT. Файлът включва правила и стилове за форматирането на различните елементи на уеб браузъра, както и шрифтове, цветове, разположение на обекти и други аспекти на дизайна, допринасящи за добрата визуализация на сайта.

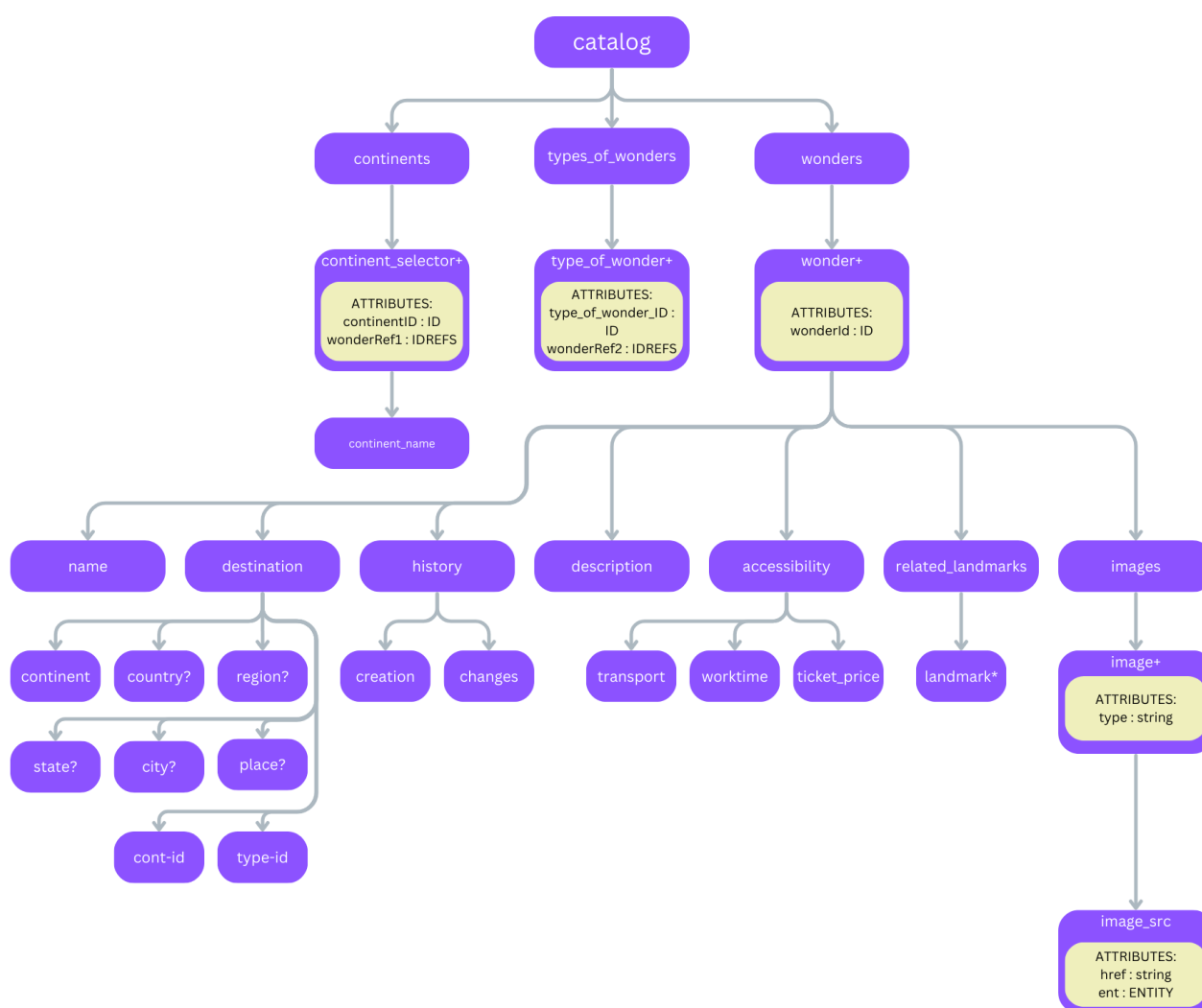
### 2.3.2 Графични изображения

За осъществяването на проекта са използвани общо 21 изображения. Намират се в папка `/images` в главната директория. Папката съдържа три подпапки - `natural_wonders`, `new_wonders`, `old_wonders` - във всяка една от тях има по 7 снимки на най-известните седем чудеса от дадената категория. Наименованията на файловете изображения се придържат към конвенцията `name_of_wonder1.jpg` като първо се поставя наименованието на чудото (ако е съставено от няколко думи, то следва думите да се отделят една от друга чрез долна черта) и числото едно след наименованието - това е направено с цел да се добавят още изображения в бъдеще за всяко едно чудо, които да следват формата на записване с име и последваща номерация, съответно 2, 3 и т.н.

## 3. Дизайн

### 3.1. Изпълнение на условията

Първата стъпка при започването на проекта беше да направим проучване за надеждна информация относно чудесата на света от различни статии, форуми на пътували до съответните чудеса, официалните сайтове на тези, които имаха такъв, да я систематизираме и да изготвим предварителен вид на XML документа, тоест прототип на архитектурата. В последствие прототипът претърпя промени, а крайната версия изглежда по този начин:



**Фиг.3 Кореново дърво на XML документа**

Диаграмата представя финалния вид на кореновото дърво на XML документа, като лилавите кутии са елементите, жълтите кутии са атрибутите на елементите, в чиито кутии са поместени, а стрелките показват, че соченият елемент е поделемента на сочещият. Добавени са също маркери за брой повторения на елементите, където + е един или повече пъти, \* е 0 или повече пъти, ? е един или нула пъти, а липсата на маркер показва, че елементът се среща точно веднъж.

Относно информацията, която сме избрали – решихме каталогът да представя подробна информация за чудесата на света по континенти и по вид чудо – природно, от старите или от новите чудеса на света. Въпреки това тя е систематична и не представя огромни количества данни за обектите, а само ключовите знания за тях. Тя е с размер средно между резюме и отделна статия за чудо.

Втори компонент е представянето на графичната информация. Това се случва чрез използването на DTD Entities. В декларацията на DTD създаваме нотация, която да бъде за снимки

с разширение .jpg и я кръщаваме jpg. След това използваме декларираната нотация, за да опишем формата на entity за всяко едно изображение, което трябва да бъде представено в страницата.

```
<!DOCTYPE catalog [  
  <!NOTATION jpg SYSTEM "image/jpeg">  
  <!ENTITY statue_of_christ_the_redeemer1 SYSTEM "images/new_wonders/statue_of_christ_the_redeemer1.jpg" NDATA jpg>  
  <!ENTITY great_wall_of_china1 SYSTEM "images/new_wonders/great_wall_of_china1.jpg" NDATA jpg>  
  <!ENTITY petra1 SYSTEM "images/new_wonders/petra1.jpg" NDATA jpg>  
  <!ENTITY chicken_itza1 SYSTEM "images/new_wonders/chicken_itza1.jpg" NDATA jpg>  
  <!ENTITY machu_picchu1 SYSTEM "images/new_wonders/machu_picchu1.jpg" NDATA jpg>  
  <!ENTITY taj_mahal1 SYSTEM "images/new_wonders/taj_mahal1.jpg" NDATA jpg>  
  <!ENTITY colosseum1 SYSTEM "images/new_wonders/colosseum1.jpg" NDATA jpg>  
  
  <!ENTITY grand_canyon1 SYSTEM "images/natural_wonders/grand_canyon1.jpg" NDATA jpg>  
  <!ENTITY northern_lights1 SYSTEM "images/natural_wonders/northern_lights1.jpg" NDATA jpg>  
  <!ENTITY mount_everest1 SYSTEM "images/natural_wonders/mount_everest1.jpg" NDATA jpg>  
  <!ENTITY paricutin1 SYSTEM "images/natural_wonders/paricutin1.jpg" NDATA jpg>  
  <!ENTITY harbour_of_rio_de_janeiro1 SYSTEM "images/natural_wonders/harbour_of_rio_de_janeiro1.jpg" NDATA jpg>  
  <!ENTITY victoria_falls1 SYSTEM "images/natural_wonders/victoria_falls1.jpg" NDATA jpg>  
  <!ENTITY great_barrier_reef1 SYSTEM "images/natural_wonders/great_barrier_reef1.jpg" NDATA jpg>  
  
  <!ENTITY hanging_gardens_of_babylon1 SYSTEM "images/old_wonders/hanging_gardens_of_babylon1.jpg" NDATA jpg>  
  <!ENTITY statue_of_zeus_at_olympia1 SYSTEM "images/old_wonders/statue_of_zeus_at_olympia1.jpg" NDATA jpg>  
  <!ENTITY colossus_of_rhodes1 SYSTEM "images/old_wonders/colossus_of_rhodes1.jpg" NDATA jpg>  
  <!ENTITY temple_of_artemis1 SYSTEM "images/old_wonders/temple_of_artemis1.jpg" NDATA jpg>  
  <!ENTITY mausoleum_at_halicarnassus1 SYSTEM "images/old_wonders/mausoleum_at_halicarnassus1.jpg" NDATA jpg>  
  <!ENTITY lighthouse_of_alexandria1 SYSTEM "images/old_wonders/lighthouse_of_alexandria1.jpg" NDATA jpg>  
  <!ENTITY pyramid_giza1 SYSTEM "images/old_wonders/pyramid_giza1.jpg" NDATA jpg>  
>]
```

Съответните entities номерираме, за да можем да добавим повече снимки при необходимост – за целите на проекта е използвано по едно изображение на чудо.

В структурата на XML документа това е отразено в елементите images, image, който може да бъде повторен многократно, ако имаме повече от една снимка и image\_src, където в атрибута ent се посочва името на entity, декларирано във вътрешното DTD на документа, а href е атрибут, който използваме само при тестването с Mozilla Firefox (повече информация в 4.3). Допълнително в .xsl файла имаме използвана функцията unparsed-entity-uri(), за извличане на информацията за местоположението на снимката от единицата(entity).

```
<div class="card_face card_face--front">  
    
</div>
```

На различни места в документа използваме ID, IDREF и IDREFS. ID имаме при елемента continent\_selector – continentID, при type\_of\_wonder - type\_of\_wonder\_ID и при всяко едно чудо wonderId. При continent\_selector и type\_of\_wonder имаме също и wonderRef1, wonderRef2, които са IDREFS, представляващи стринг от wonderId, използвани за сортировките по континенти и по тип. Спазили сме условието и „Връзките в каталога – напр. между групи и подгрупи, между райони и дестинации, и др. – да се описват чрез (съставни) ключове и референции към тях. “

```

<xs:key name="wonderKey">
  <xs:selector xpath="catalog/wonders/wonder"/>
  <xs:field xpath="@wonderId"/>
  <xs:field xpath="name"/>
</xs:key>

<xs:key name="continentKey">
  <xs:selector xpath="catalog/continents/continent_selector"/>
  <xs:field xpath="@continentID"/>
</xs:key>


<xs:key name="typeKey">
  <xs:selector xpath="catalog/types_of_wonders/type_of_wonder"/>
  <xs:field xpath="@type_of_wonder_ID"/>
</xs:key>

<xs:keyref name="wonderContinentKeyRef" refer="continentKey">
  <xs:selector xpath="catalog/wonders/wonder" />
  <xs:field xpath="destination/cont-id"/>
</xs:keyref>

<xs:keyref name="wonderTypeKeyRef" refer="typeKey">
  <xs:selector xpath="catalog/wonders/wonder" />
  <xs:field xpath="destination/type-id"/>
</xs:keyref>

```

За всяко от чудесата на света се поддържа структурирана информация - име, псевдоними, местоположение (място, град, държава, континент, регион и щат), история (създаване и промени през годините), описание на чудото, снимки, достъпност (как се стига до обекта – с какъв транспорт, работно време и цена за посещение). Също така съдържа информация за свързани интересни обекти и забележителности. За всеки тип чудо – природно, на стария или на новия свят сме дали по 7 примера.



**Име**  
Храмът на Зевс в Олимпия

**Дестинация**  
Европа; Гърция; Олимпия; Археологически парк Олимпия

**Работно време**  
Работното време на археологическия парк е от 8:00 до 20:00 часа през лятото и от 8:00 до 17:00 часа през зимата.

**Описание**  
Храмът на Зевс е бил един от най-големите и впечатляващи храмове на Древна Гърция. Той е имал размери от 64 метра на 27,5 метра и е бил издигнат на три стъпала. Имал е шест колони на всяка къса страна и тринадесет колони на всяка дълга страна, които са поддържали дървен покрив, покрит с мраморни плочи. Във вътрешността на храма се е намирала огромна статуя на Зевс от злато и слоновя кост, която е била едно от седемте чудеса на света. Статуята е била висока 13 метра и е изобразявала Зевс, седнал на трон, държащ скиптер и победна статуетка на Нике. Тя била изработена от известния скулптор Фидий, който е работил в специална работилница в Олимпия. Храмът е бил украсен с метопи и фризове, изобразяващи сцени от митологията и олимпийските игри.

### История на създаването

Храмът на Зевс е построен в чест на главния бог на древните гърци, който е покровител на Олимпийските игри. Той е финансиран от полиса Елис. Архитект на храма е Либон, който е използвал дорически стил и местен камък за строежа. Храмът е завършен около 457 г. пр. Хр. и е считан за един от най-величествените и хармонични храмове на Древна Гърция.

### Промени през годините

Храмът на Зевс е бил разрушен от земетресения, пожари и набези през вековете. През 426 г. християнският император Теодосий II е заповядал да се разруши храмът и да се премахне култът към Зевс. През 522 и 551 г. земетресения са съборили колоните и покрива на храма, а през 1829 г. френски археолози са откраднали неговите мраморни скулптури. Днес от храма са останали само развалини и фрагменти от декорацията му.

### Достъпност

Храмът на Зевс се намира в археологическия парк Олимпия, който е на около 300 км западно от Атина. Може да се стигне до парка с автобус, влак или кола. Автобусите тръгват от автогарата Кифисос в Атина и отнемат около 4-5 часа. Влаковете тръгват от централната гара в Атина и отнемат около 5-6 часа. Колата е по-удобен вариант, тъй като позволява да се посетят и други забележителности по пътя като Делфи, Нафплио и Микена.

### Цени на билети

Цената на билета за вход в парка е 12 евро за възрастни и 6 евро за деца, студенти и пенсионери. Билетът включва и вход в археологическия музей на Олимпия, където се пазят много ценни находки от храма и други паметници.

### Свързани чудеса

- Стадионът на Олимпия - това е мястото, където са се провеждали първите Олимпийски игри през VIII век пр. Хр. Тук са се състезавали атлети в различни дисциплини, като бягане, борба, дисково хвърляне и копиехвърляне. Стадионът е имал капацитет за 45 000 зрители и е бил достъпен само за мъже.
- Храмът на Хера - това е най-старият храм в Олимпия, построен около 600 г. пр. Хр. в чест на Хера, съпругата на Зевс. Тук се е намирала статуята на Хера, също дело на Фидий. Също така тук се е запалвал олимпийският огън.
- Музеят на Олимпийските игри - това е съвременен музей, който е посветен на историята и духа на Олимпийските игри. Тук се пазят много предмети, свързани с игрите, като медали, факли, плакати, костюми и други. Често се организират изложби, семинари и образователни програми за деца и възрастни.

```
<!ELEMENT catalog (continents, types_of_wonders, wonders)>
<!ELEMENT continents (continent_selector+)>
<!ELEMENT types_of_wonders (type_of_wonder+)>
<!ELEMENT wonders (wonder+)>
<!ELEMENT continent_selector (continent_name)>
<!ELEMENT type_of_wonder (type_of_wonder_name)>
<!ELEMENT continent_name (#PCDATA)>
<!ELEMENT type_of_wonder_name (#PCDATA)>
<!ELEMENT wonder (name, nickname?, destination, history, description, accessibility, related_landmarks, images)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT nickname (#PCDATA)>
<!ELEMENT destination (continent?, country?, region?, state?, city?, place?, cont-id, type-id)>
<!ELEMENT history (creation, changes)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT accessibility (transport, worktime, ticket_price)>
<!ELEMENT related_landmarks (landmark*)>
<!ELEMENT images (image+)>
<!ELEMENT continent (#PCDATA)>
<!ELEMENT region (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT place (#PCDATA)>
<!ELEMENT cont-id (#PCDATA)>
<!ELEMENT type-id (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT creation (#PCDATA)>
<!ELEMENT changes (#PCDATA)>
<!ELEMENT transport (#PCDATA)>
<!ELEMENT worktime (#PCDATA)>
<!ELEMENT ticket_price (#PCDATA)>
<!ELEMENT landmark (#PCDATA)>
<!ELEMENT image (image_src)>
<!ELEMENT image_src (#PCDATA)>
<!ATTLIST catalog xmlns:xsi CDATA #FIXED "http://www.w3.org/2001/XMLSchema-instance">
<!ATTLIST catalog xsi:namespaceSchemaLocation CDATA #FIXED "wonders_of_the_world.xsd">
<!ATTLIST continent_selector continentID ID #REQUIRED>
<!ATTLIST continent_selector wonderRef1 IDREFS #IMPLIED>
<!ATTLIST type_of_wonder type_of_wonder_ID ID #REQUIRED>
<!ATTLIST type_of_wonder wonderRef2 IDREFS #IMPLIED>
<!ATTLIST wonder wonderId ID #REQUIRED>
<!ATTLIST image type NMTOKEN #REQUIRED>
<!ATTLIST image_src href CDATA #REQUIRED>
<!ATTLIST image_src ent ENTITY #REQUIRED>
```



Валидирането на документа се извършва посредством вътрешно DTD и XMLSchema. Името на файла е wonders\_of\_the\_world.xsd.

```
<xs:complexType name="wonder_type">
  <xs:sequence>
    <xs:element name="wonder" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string" />
          <xs:element name="nickname" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="destination">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="continent" type="xs:string" minOccurs="1"/>
                <xs:element name="country" type="xs:string" minOccurs="0"/>
                <xs:element name="region" type="xs:string" minOccurs="0"/>
                <xs:element name="state" type="xs:string" minOccurs="0"/>
                <xs:element name="city" type="xs:string" minOccurs="0"/>
                <xs:element name="place" type="xs:string" minOccurs="0"/>
                <xs:element name="cont-id" type="xs:IDREF"/>
                <xs:element name="type-id" type="xs:IDREF"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="history">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="creation" type="xs:string" />
                <xs:element name="changes" type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="description" type="xs:string" />
          <xs:element name="accessibility">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="transport" type="xs:string" />
                <xs:element name="worktime" type="xs:string" />
                <xs:element name="ticket_price" type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Създаденото XML съдържание представяме, като го трансформираме чрез XSLT до HTML и чрез отделен CSS файл добавяме стила на представяне на информацията. Сортирането на представените обекти се извършва в wonders\_of\_the\_world.xsl с възможност за сортиране на представените обекти по структурата на XML документа (стандартен вид), по континенти, по тип на чудото и по азбучен ред. За целта използваме елементите continents и types\_of\_wonders от XML документа, бутони за селектиране на избора ни в HTML, темплейти за визуализиране на информацията от чудо и функция за разделяне на низа от IDREFS за всяка категория. Също така използваме и xsl:sort за подреждане по азбучен ред.

```
<continents>
  <continent_selector continentID = "AS" wonderRef1="PETRA GREAT_WALL TAJ_MAHAL EVEREST HANGING_GARDENS_OF_BABYLON TEMPLE_OF_ARTEMIS_AT_EPHEBUS">
    <continent_name>Азия</continent_name>
  </continent_selector>
</continents>
```

## Каталог на Чудесата на света

Сортиране по структурата на XML

Сортиране по континенти

Сортиране по тип на чудо

Сортиране по азбучен ред

### Азия :



### Австралия :



#### Име

Големият бариерен риф  
Най-големият жив организъм на Земята

#### Дестинация

Австралия и Океания; Австралия; Куинсланд; Коралово море

#### Работно време

Няма ограничения за посещение на рифа, но някои от туристическите оператори и доставчиците на услуги могат да имат различни работни

## Европа :

Сортиране по структурата на XML

Сортиране по континенти

Сортиране по тип на чудо

Сортиране по азбучен ред

### Природни чудеса :



#### Име

Гранд Каньон  
Чудото на природата

#### Дестинация

Северна Америка; САЩ; Аризона; Национален парк Гранд Каньон

#### Работно време

24 часа в денонощието, цялостно

### Стари чудеса на света :



#### Име

Висящите градини на Вавилон  
Висящите градини на Семирамида

#### Дестинация

Азия; Ирак; Вавилон;

#### Работно време


NONE

#### Сортиране



## Каталог на Чудесата на света

Сортиране по структурата на XML
Сортиране по континенти
Сортиране по тип на чудо
Сортиране по азбучен ред



**Име**  
Александрийският фар

**Дестинация**  
Африка; Древен Египет; град Александрия; остров Фарос

**Работно време**



**Име**  
Великата китайска стена  
Стената; Стената на дракона; Стена, дълга 10000 ли; Чанчън

**Дестинация**  
Азия; Китай;

**Работно време**  
Бадалинг - Работното време на този раздел е от 6:30 до 19:00 часа през лятото и от 7:00 до 18:00 часа през зимата. Цзиншанлинг - Работното време на този раздел е от 8:00 до 17:00 часа през цялата година.

**Описание**  
Великата китайска стена е огромно съоръжение, което пресича северната и централната част на Китай. Тя има много различни секции, които са построени в различни периоди от историята. Изградена е от различни материали, като камък, тухли, дърво и пръст, и има много наблюдателни кули, казарми, гарнизони и системи за сигнализация. Символ е на китайската цивилизация и история.

```
<div id="4">
  <xsl:apply-templates select="/catalog/wonders/wonder">
    <xsl:sort select="name" order="ascending"/>
  </xsl:apply-templates>
</div>
```

## 3.2. Архитектура на решението и обяснения по кода

Първите две технологии, които сме използвали, са XML и XMLSchema. XML използваме, за да направим конструкцията на предварително селектираната информация, а XMLSchema за валидирани на документа ни и дефиниране на неговата структура. Кореновият елемент на проекта е catalog, той съдържа три поделемента – continents, types\_of\_wonders и wonders.

```

<xs:element name="catalog">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="continents" type="continents_type"/>
      <xs:element name="types_of_wonders" type="types_of_wonders_type"/>
      <xs:element name="wonders" type="wonder_type"/>
    </xs:sequence>
  </xs:complexType>

  <xs:key name="wonderKey">
    <xs:selector xpath="catalog/wonders/wonder"/>
    <xs:field xpath="@wonderID"/>
    <xs:field xpath="name"/>
  </xs:key>

  <xs:key name="continentKey">
    <xs:selector xpath="catalog/continents/continent_selector"/>
    <xs:field xpath="@continentID"/>
  </xs:key>

  <xs:key name="typeKey">
    <xs:selector xpath="catalog/types_of_wonders/type_of_wonder"/>
    <xs:field xpath="@type_of_wonder_ID"/>
  </xs:key>

  <xs:keyref name="wonderContinentKeyRef" refer="continentKey">
    <xs:selector xpath="catalog/wonders/wonder" />
    <xs:field xpath="destination/cont-id"/>
  </xs:keyref>

  <xs:keyref name="wonderTypeKeyRef" refer="typeKey">
    <xs:selector xpath="catalog/wonders/wonder" />
    <xs:field xpath="destination/type-id"/>
  </xs:keyref>
</xs:element>

```

В continents елемента са разположени 7 continent\_selector елемента, чиято цел е първо да дефинират 7-те континента на планетата и второ – да служат като ключ, който да бъде рефериран от чудесата за осъществяването на връзката между едно чудо и континента, на който се намира. В този елемент имаме два атрибута – continentID, използван за дефинирането на ключа, тоест за идентификатор на континента, и wonderRef1, който е IDREFS, чиято цел е да посочи идентификаторите на чудесата, които се намират в този континент, разделени от интервал.

В types\_of\_wonders елемента са разположени type\_of\_wonder елемента, чиято цел е да дефинират различни типове чудеса – природни, стари чудеса на света и нови чудеса на света и второ – да служат като ключ, който да бъде рефериран от чудесата за осъществяването на връзката между едно чудо и неговия тип. В този елемент имаме два атрибута – type\_of\_wonder\_ID, използван за

дефинирането на ключа, тоест за идентификатор на типа, и wonderRef2, който е IDREFS, чиято цел е да посочи идентификаторите на чудесата, които са от този тип, разделени от интервал.

```
<xs:complexType name="continents_type">
  <xs:sequence>
    <xs:element name="continent_selector" maxOccurs="7">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="continent_name" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="continentID" type="xs:ID" use="required"/>
        <xs:attribute name="wonderRef1" type="xs:IDREFS" use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="types_of_wonders_type">
  <xs:sequence>
    <xs:element name="type_of_wonder" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type_of_wonder_name" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="type_of_wonder_ID" type="xs:ID" use="required"/>
        <xs:attribute name="wonderRef2" type="xs:IDREFS" use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```
<continents>
  <continent_selector continentID = "AS" wonderRef1="PETRA GREAT_WALL TAJ_MAHAL EVEREST HANGING_GARDENS_OF_BABYLON TEMPLE_OF_ARTEMIS_AT_EPHEBUS">
    <continent_name>Азия</continent_name>
  </continent_selector>

  <continent_selector continentID = "AU" wonderRef1="GREAT_BARRIER_REEF">
    <continent_name>Австралия</continent_name>
  </continent_selector>

  <continent_selector continentID = "EU" wonderRef1="COLLOSEUM STATUE_OF_ZEUS_AT_OLYMPIA COLOSSUS_OF_RHODES MAUSOLEUM_AT_HALICARNASSUS">
    <continent_name>Европа</continent_name>
  </continent_selector>

  <continent_selector continentID = "SA" wonderRef1="CHICHEN_ITZA STATUE_CHRIST MACHU_PICCHU RIO_DE_JANEIRO_HARBOUR">
    <continent_name>Южна Америка</continent_name>
  </continent_selector>

  <continent_selector continentID = "NA" wonderRef1="GRAND_CANYON PARICUTIN">
    <continent_name>Северна Америка</continent_name>
  </continent_selector>

  <continent_selector continentID = "AF" wonderRef1="VICTORIA_FALLS GREAT_PYRAMID_OF_GIZA Lighthouse_OF_ALEXANDRIA">
    <continent_name>Африка</continent_name>
  </continent_selector>

  <continent_selector continentID = "AN" wonderRef1="AURORA">
    <continent_name>Антарктида</continent_name>
  </continent_selector>
</continents>
```

```

<types_of_wonders>
  <type_of_wonder type_of_wonder_ID = "OW" wonderRef2="HANGING_GARDENS_OF_BABYLON STATUE_OF_ZEUS_AT_OLYMPIA TEMPLE_OF_ARTEMIS_AT_EPHEBUS MAUSOLEUM_AT_HALICARNASSUS
    LIGHTHOUSE_OF_ALEXANDRIA COLOSSUS_OF_RHODES GREAT_PYRAMID_OF_GIZA">
    <type_of_wonder_name>Стари чудеса на света</type_of_wonder_name>
  </type_of_wonder>

  <type_of_wonder type_of_wonder_ID = "NW" wonderRef2="STATUE_CHRIST GREAT_WALL PETRA CHICHEN_ITZA MACHU_PICCHU TAJ_MAHAL COLLOSEUM">
    <type_of_wonder_name>Нови чудеса на света</type_of_wonder_name>
  </type_of_wonder>

  <type_of_wonder type_of_wonder_ID = "NAW" wonderRef2="GRAND_CANYON AURORA EVEREST PARICUTIN RIO_DE_JANEIRO_HARBOUR VICTORIA_FALLS GREAT_BARRIER_REEF ">
    <type_of_wonder_name>Природни чудеса</type_of_wonder_name>
  </type_of_wonder>
</types_of_wonders>

```

Следващият елемент е wonders, който съдържа конкретната информация за чудесата в своите поделементи wonder. В него имаме атрибут wonderId, който е ID и дава уникален идентификатор на чудото, а посредством този атрибут и името на чудото се формира съставен ключ, който служи за валидиране на уникалността на всяко чудо.

```

</xs:sequence>
  <xs:attribute name="wonderId" type="xs:ID" use="required"/>
</xs:complexType>

```

```

<xs:element name="wonder" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="nickname" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="destination">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="continent" type="xs:string" minOccurs="1"/>
            <xs:element name="country" type="xs:string" minOccurs="0"/>
            <xs:element name="region" type="xs:string" minOccurs="0"/>
            <xs:element name="state" type="xs:string" minOccurs="0"/>
            <xs:element name="city" type="xs:string" minOccurs="0"/>
            <xs:element name="place" type="xs:string" minOccurs="0"/>
            <xs:element name="cont-id" type="xs:IDREF"/>
            <xs:element name="type-id" type="xs:IDREF"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="history">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="creation" type="xs:string" />
            <xs:element name="changes" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

В структурата на wonder сме определили поделелементите да имат точен ред на изписване, където name – името на чудото, nickname – псевдонима му, destination - местоположението, history – исторически бележки, accessibility – достъпността, related\_landmarks -свързаните чудеса и images – снимките се срещат точно по един път. В местоположението всички поделелементи освен continent, cont-id и type-id могат да присъстват или не, но тези три са задължителни, тъй като трябва да знаем поне на кой континент се намира чудото, а cont-id и type-id реферират ключовете на континентите и типовете чудеса за създаване на отчетливи връзки в документа. Историческите бележки имат два поделелемента – creation и changes, които представят текстова информация за създаването на чудото и промени, които е претърпяло през годините като например реконструкции.

Продължавайки, имаме поделелемента description, който съдържа кратка информация за чудото, а в accessibility имаме три поделелемента – transport, worktime и ticket\_price, които описват съответно транспорта (как се стига до чудото), работното време и цената на билет за достъп. В related\_landmarks са посочени кратки данни за близки по местоположение други забележителности като всяка една е отделена в самостоятелен елемент landmark, който може да се повтаря от 0 до безброй пъти.

```
<xs:element name="description" type="xs:string" />
<xs:element name="accessibility">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="transport" type="xs:string" />
      <xs:element name="worktime" type="xs:string" />
      <xs:element name="ticket_price" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="related_landmarks">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="landmark" type="xs:string" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Най-интересната част са графичните елементи, като те са обособени в елемента images. Негов поделелемент е image, повторен поне веднъж, тоест може да имаме повече от една снимка, но в рамките на проекта сме се ограничили само до една. Елементът image\_src съдържа два атрибута – href и ent. Тяхната роля е да се посочи кое изображение да се намери за това чудо. Ent се използва ексклузивно във всички файлове – то е връзката с дефинираните единици за изображения, а href е резервен, който дава директна връзка към местоположението на jpg файла, за да може при желание да се визуализират снимките в браузъра Mozilla Firefox.

В XML документа имаме вътрешно DTD, което показва правилата за структурата както е дефинирана в XMLSchema дефиницията.

```

<xs:element name="images">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="image" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="image_src">
              <xs:complexType>
                <xs:attribute name="ent" type="xs:ENTITY" use="required"/>
                <xs:attribute name="href" type="xs:string" use="required"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="type" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!DOCTYPE catalog [
  <!NOTATION jpeg SYSTEM "image/jpeg">
  <!ENTITY statue_of_christ_the_redeemer1 SYSTEM "images/new_wonders/statue_of_christ_the_redeemer1.jpg" NDATA jpeg>
  <!ENTITY great_wall_of_china1 SYSTEM "images/new_wonders/great_wall_of_china1.jpg" NDATA jpeg>
  <!ENTITY petra1 SYSTEM "images/new_wonders/petra1.jpg" NDATA jpeg>
  <!ENTITY chichen_itza1 SYSTEM "images/new_wonders/chichen_itza1.jpg" NDATA jpeg>
  <!ENTITY machu_picchu1 SYSTEM "images/new_wonders/machu_picchu1.jpg" NDATA jpeg>
  <!ENTITY taj_mahal1 SYSTEM "images/new_wonders/taj_mahal1.jpg" NDATA jpeg>
  <!ENTITY colosseum1 SYSTEM "images/new_wonders/colosseum1.jpg" NDATA jpeg>

  <!ENTITY grand_canyon1 SYSTEM "images/natural_wonders/grand_canyon1.jpg" NDATA jpeg>
  <!ENTITY northern_lights1 SYSTEM "images/natural_wonders/northern_lights1.jpg" NDATA jpeg>
  <!ENTITY mount_everest1 SYSTEM "images/natural_wonders/mount_everest1.jpg" NDATA jpeg>
  <!ENTITY paricutin1 SYSTEM "images/natural_wonders/paricutin1.jpg" NDATA jpeg>
  <!ENTITY harbour_of_rio_de_janeiro1 SYSTEM "images/natural_wonders/harbour_of_rio_de_janeiro1.jpg" NDATA jpeg>
  <!ENTITY victoria_falls1 SYSTEM "images/natural_wonders/victoria_falls1.jpg" NDATA jpeg>
  <!ENTITY great_barrier_reef1 SYSTEM "images/natural_wonders/great_barrier_reef1.jpg" NDATA jpeg>

  <!ENTITY hanging_gardens_of_babylon1 SYSTEM "images/old_wonders/hanging_gardens_of_babylon1.jpg" NDATA jpeg>
  <!ENTITY statue_of_zeus_at_olympia1 SYSTEM "images/old_wonders/statue_of_zeus_at_olympia1.jpg" NDATA jpeg>
  <!ENTITY colossus_of_rhodes1 SYSTEM "images/old_wonders/colossus_of_rhodes1.jpg" NDATA jpeg>
  <!ENTITY temple_of_artemis1 SYSTEM "images/old_wonders/temple_of_artemis1.jpg" NDATA jpeg>
  <!ENTITY mausoleum_at_halicarnassus1 SYSTEM "images/old_wonders/mausoleum_at_halicarnassus1.jpg" NDATA jpeg>
  <!ENTITY lighthouse_of_alexandria1 SYSTEM "images/old_wonders/lighthouse_of_alexandria1.jpg" NDATA jpeg>
  <!ENTITY pyramid_giza1 SYSTEM "images/old_wonders/pyramid_giza1.jpg" NDATA jpeg>

```

Следващите технологии, които трябваше да използваме, бяха XSLT, HTML и CSS. Те са свързани с трансформирането на XML документа до HTML и визуализирането му посредством похватите на HTML и CSS.

```

<xsl:template match="wonder">
  <div class="scene_container">
    <div class="scene">
      <div class="card">
        <div class="card__face card__face--front">
          
          <!--Този ред се добавя към img в края, ако искате да се визуализират изображения
          на Mozilla Firefox, тъй като layout engine-ът на браузъра не поддържа ХРАТН
          метода unparsed-entity-uri()-->
          <!--onerror="this.src='{images/image/image_src/@href}';"-->

          <div class="card_text_container">
            <div class="card_text">
              <p>Име</p>
              <xsl:value-of select="name"/>
              <br/>
              <xsl:value-of select="nickname"/>
              <p>Дестинация</p>
              <xsl:if test="destination/continent">
                <xsl:value-of select="concat(destination/continent, ' ')" />
              </xsl:if>
              <xsl:if test="destination/country">
                <xsl:value-of select="concat(destination/country, ' ')" />
              </xsl:if>
              <xsl:if test="destination/region">
                <xsl:value-of select="concat(destination/region, ' ')" />
              </xsl:if>
              <xsl:if test="destination/state">
                <xsl:value-of select="concat(destination/state, ' ')" />
              </xsl:if>
              <xsl:if test="destination/city">
                <xsl:value-of select="concat(destination/city, ' ')" />
              </xsl:if>
              <xsl:if test="destination/place">
                <xsl:value-of select="destination/place"/>
              </xsl:if>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</xsl:template>

```

```

      </xsl:if>
      <p>Работно време</p>
      <xsl:value-of select="accessibility/worktime"/>
      <p>Описание</p>
      <xsl:value-of select="description"/>
    </div>
  </div>
  <div class="card__face card__face--back">
    <div class="card__text_container_back">
      <div class="card__text_back_1st_col">
        <p>История на създаването</p>
        <xsl:value-of select="history/creation"/>
        <p>Промени през годините</p>
        <xsl:value-of select="history/changes"/>
        <p>Достъпност</p>
        <xsl:value-of select="accessibility/transport"/>
      </div>
      <div class="card__text_back_2nd_col">
        <p>Цени на билети</p>
        <xsl:value-of select="accessibility/ticket_price"/>
        <p>Свързани чудеса</p>
        <xsl:for-each select="related_landmarks/landmark">
          <ul>
            <li><xsl:value-of select="."/></li>
          </ul>
        </xsl:for-each>
      </div>
    </div>
  </div>
</div>
</div>
</xsl:template>

```

Следният темплейт извлича информацията от XML документа за едно чудо и я разпределя в съответните div класове. Той се използва на много места в документа. Важни моменти, които трябва да отбележим, са:

- match="wonder", с което казваме да се търси информацията в елемента wonder.
- С xsl:if конструкцията проверяваме поотделно за съществуването на всеки един от поделементите на destination и при наличието му, го добавяме с ; накрая.
- За related\_landmarks/landmark използваме xsl:for-each, за да обходим и изведем всеки намерен текст в съответен елемент като списък.



Другият темплейт, който сме създали, работи като функция, която взима wonderRef1 или wonderRef2 и извлича от техните стрингове всички посочени чудеса. Работи по следния начин – взима подадения му като параметър низ и ако той съдържа някакъв текст и интервал, взима текста преди интервала и използвайки темплейта за чудо, подбира този, който за стойност на атрибута си има стойността, която е извадена от низа. След това подава рекурсивно низа, който е след интервала и извършва многократно същите действия. Когато достигне последния идентификатор на чудо, който няма интервал след него, преминаваме в otherwise случая, където използвайки темплейта за чудо, подбира този, който за стойност на атрибута си има стойността на последния идентификатор от низа и прекратява изпълнението си.

```
<xsl:template name="splitString">
  <xsl:param name="string"/>
  <xsl:choose>
    <xsl:when test="contains($string, ' ')">
      <xsl:if test="$string">
        <xsl:variable name="wonder" select="substring-before($string, ' ')" />
        <xsl:apply-templates select="/catalog/wonders/wonder[@wonderId=$wonder]" />

        <xsl:call-template name="splitString">
          <xsl:with-param name="string" select="substring-after($string, ' ')" />
        </xsl:call-template>
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name="wonder" select="$string" />
      <xsl:apply-templates select="/catalog/wonders/wonder[@wonderId=$wonder]" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

След това имаме основния темплейт в документа, който съдържа html-a. В head определяме връзка с външен файл от тип text/css, където се съдържа стилистичната информация. В скрипт има три JavaScript функции, чиито цели ще опишем:

- function showDiv(divId) се използва като действие за бутоните. Когато бъде натиснат бутон се подава съответен идентификатор на div клас, който да бъде показан, а останалите биват скрити.
- window.onload = function() дава инструкции, че по подразбиране при отваряне на XML документа, ще се покаже сортиран по структурата си.
- function toggleFooter() премества футъра нагоре и надолу при натискане на показания бутон, за да покаже информацията в него – данните на изготвилите го студенти.



```

<xsl:template match="/">
  <html>
    <head>
      <link rel="stylesheet" type="text/css" href="wonders_of_the_world.css"/>
      <script>
        function showDiv(divId) {
          var divToShow = document.getElementById(divId);
          var otherDivs = ['1', '2', '3', '4'].filter(item => item !== divId);

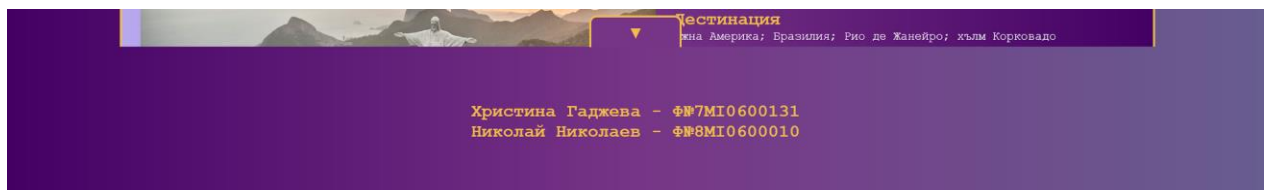
          divToShow.style.display = 'block';
          otherDivs.forEach(otherDivId => {
            var otherDiv = document.getElementById(otherDivId);
            otherDiv.style.display = 'none';
          });
        }

        window.onload = function() {
          showDiv('1');
        };

        function toggleFooter() {
          const footer = document.getElementById('footer');
          const toggleButton = document.querySelector('.toggle-button');
          const toggleButtonContainer = document.querySelector('.toggle-button-container');

          if (footer.style.bottom === '0px') {
            footer.style.bottom = '-205px';
            toggleButtonContainer.style.bottom = '0px';
            toggleButton.innerHTML = '▲';
          } else {
            footer.style.bottom = '0px';
            toggleButtonContainer.style.bottom = '202px';
            toggleButton.innerHTML = '▼';
          }
        }
      </script>
    </head>

```



В тялото създаваме 4 бутона – по един за всеки вид сортировка, като 1, 2, 3 и 4 са id на съответните div класове. При натискане на бутона се изпълнява горе описаната JavaScript функция showDiv(divId). По-долу определяме за краткост при последващото им използване променливи, съдържащи XPath пътища до съответните continent\_selector за всеки един континент. При първия div клас просто извикваме темплейта за чудесата и в браузъра се визуализират отделните карти с описания за всяко едно чудо по реда в XML документа. Втория div клас е за сортировката по континенти. Там първо показваме името на текущо избрания континент, като те са подредени по азбучен ред, след което извикваме темплейта splitString с параметър wonderRef1 на съответния континент, за да укажем да се покажат чудесата от неговия низ с референции. Повтаряме 7 пъти като променяме променливата (в първия случай \$Asia).

```
<body>
  <h1>Каталог на Чудесата на света</h1>

  <div class="buttons">
    <button onclick="showDiv('1')">Сортиране по структурата на XML</button>
    <button onclick="showDiv('2')">Сортиране по континенти</button>
    <button onclick="showDiv('3')">Сортиране по тип на чудо</button>
    <button onclick="showDiv('4')">Сортиране по азбучен ред</button>
  </div>

  <xsl:variable name="Asia" select="/catalog/continents/continent_selector[@continentID='AS']"/>
  <xsl:variable name="Australia" select="/catalog/continents/continent_selector[@continentID='AU']"/>
  <xsl:variable name="Antarctica" select="/catalog/continents/continent_selector[@continentID='AN']"/>
  <xsl:variable name="Afrika" select="/catalog/continents/continent_selector[@continentID='AF']"/>
  <xsl:variable name="Europe" select="/catalog/continents/continent_selector[@continentID='EU']"/>
  <xsl:variable name="North_America" select="/catalog/continents/continent_selector[@continentID='NA']"/>
  <xsl:variable name="South_America" select="/catalog/continents/continent_selector[@continentID='SA']"/>

  <div id="1">
    <xsl:apply-templates select="/catalog/wonders/wonder"/>
  </div>

  <div class="continents" id="2">
    <h3><xsl:value-of select="$Asia/continent_name"/></h3>
    <xsl:call-template name="splitString">
      <xsl:with-param name="string" select="$Asia/@wonderRef1"/>
    </xsl:call-template>

    <h3><xsl:value-of select="$Australia/continent_name"/></h3>
    <xsl:call-template name="splitString">
      <xsl:with-param name="string" select="$Australia/@wonderRef1"/>
    </xsl:call-template>

    <h3><xsl:value-of select="$Antarctica/continent_name"/></h3>
    <xsl:call-template name="splitString">
      <xsl:with-param name="string" select="$Antarctica/@wonderRef1"/>
    </xsl:call-template>
```

Третия div клас работи по същия начин като втория, но с данните от видовете чудеса (types\_of\_wonders). За четвъртия клас извикваме темплейта за принтиране на чудесата, като с xsl:sort избираме да са сортирани в лексикографска наредба по името на чудото в нарастващ ред.

```

<xsl:variable name="Natural" select="/catalog/types_of_wonders/type_of_wonder[@type_of_wonder_ID='NAW']"/>
<xsl:variable name="New" select="/catalog/types_of_wonders/type_of_wonder[@type_of_wonder_ID='NW']"/>
<xsl:variable name="Old" select="/catalog/types_of_wonders/type_of_wonder[@type_of_wonder_ID='OW']"/>

<div class="types" id="3">
  <h3><xsl:value-of select="$Natural/type_of_wonder_name"/>:</h3>
  <xsl:call-template name="splitString">
    <xsl:with-param name="string" select="$Natural/@wonderRef2"/>
  </xsl:call-template>

  <h3><xsl:value-of select="$New/type_of_wonder_name"/>:</h3>
  <xsl:call-template name="splitString">
    <xsl:with-param name="string" select="$New/@wonderRef2"/>
  </xsl:call-template>

  <h3><xsl:value-of select="$Old/type_of_wonder_name"/>:</h3>
  <xsl:call-template name="splitString">
    <xsl:with-param name="string" select="$Old/@wonderRef2"/>
  </xsl:call-template>
</div>

<div id="4">
  <xsl:apply-templates select="/catalog/wonders/wonder">
    <xsl:sort select="name" order="ascending"/>
  </xsl:apply-templates>
</div>

```

Последната част е футъра, който показва имената ни. Той се задвижва от бутон, който извиква третата JavaScript функция toggleFooter().

```

<div class="toggle-button-container">
  <div class="toggle-button" onclick="toggleFooter()">
    ▲
  </div>
</div>

<footer class="footer" id="footer">
  <div class="footer-content">
    <p>Христина Гаджева - ФН№7МИ0600131<br/>
    Николай Николаев - ФН№8МИ0600010</p>
  </div>
</footer>
</body>

```

## 4. Тестване

Извършихме тестване в 4 браузъра – Microsoft Edge, Google Chrome, Mozilla Firefox и Internet Explorer. Тъй като всеки браузър си има специфики се наложи допълнително конфигуриране и създаване на локален сървър през командния терминал на Windows 11. Проектът съдържа 4 файла – `wonders_of_the_world.xml`, `wonders_of_the_world.xsd`, `wonders_of_the_world.xsl`, `wonders_of_the_world.css`. Подробно описание на Спецификите на всеки браузър може да намерите в следните подточки:

### 4.1. Microsoft Edge и Google Chrome

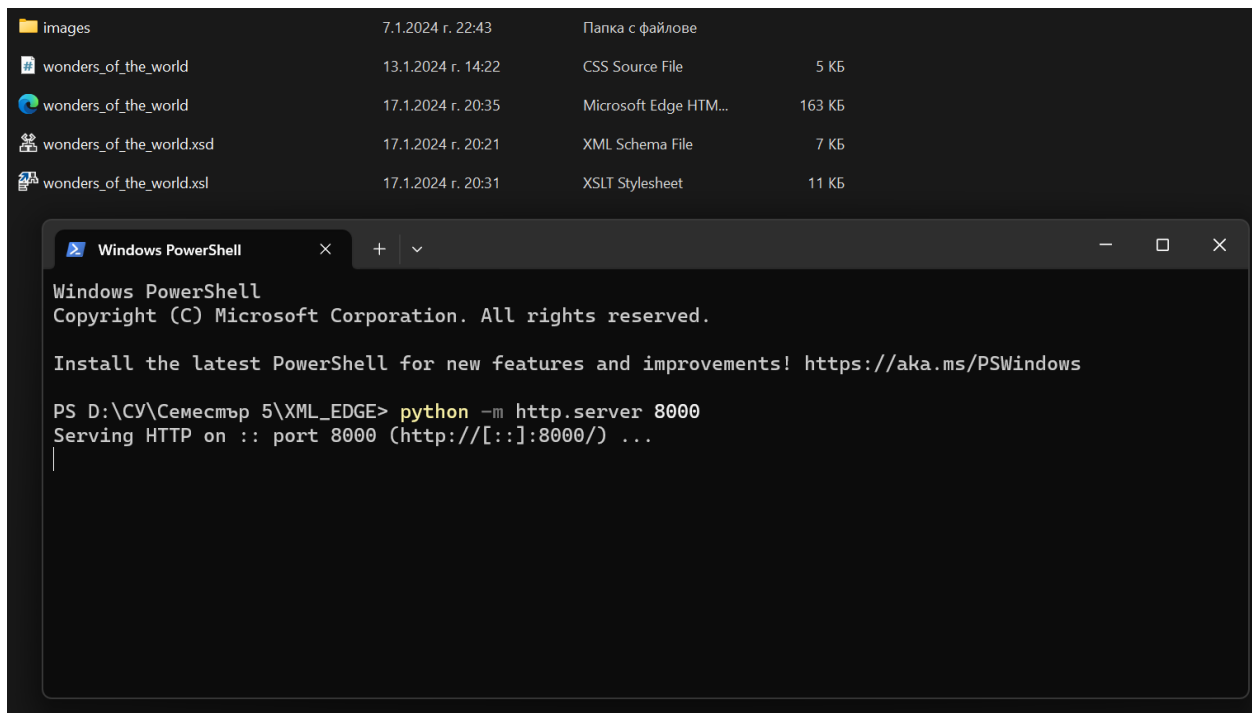
Двата браузъра Microsoft Edge и Google Chrome са базирани на Chromium, което е безплатен софтуер с отворен код, разработван основно от Google, предоставящ ядро за браузър. По тази причина те имат сходно поведение и са групирани в една категория. Важно уточнение, което трябва да направим, е, че Microsoft Edge от преди 2018 година е написан с друг код на ядрото на браузъра и там може да има разлики. В съвременното обаче почти всички компютри под Windows оперират с Chromium-базирания версия.

Единственото нещо, което се налага като конфигурация при тези два браузъра, е създаването на локален сървър през командния терминал. Това е нужно, защото защитите на двата браузъра не позволяват на файлове да достъпват други файлове от файловата система, когато са отворени в браузър. Тоест в този случай браузърът ще открие, че има такива файлове – примерно `xsl stylesheet`, но ще покаже бял екран в браузъра, защото не му е позволено да ги изпълни. Инициализирайки абсолютно обикновен локален сървър без никакви допълнителни настройки можем да видим успешно нашият обработен XML документ.

Предварителни стъпки, нужни за повтаряне на същата процедура:

1. При неинсталиран Python на компютъра – инсталирате някоя от последните версии на Python 3.9, 3.10, 3.11, 3.12 са сигурен вариант от уебсайта на Python <https://www.python.org/>. При инсталацията се следват инструкциите на инсталатора и се отбелязва опцията за добавяне на Python към PATH.
2. Ако имате инсталиран Python или сте изпълнили първа стъпка, трябва да намерите местоположението на папката, в която се намират файловете от проекта. В нея стартирате командния терминал на Windows (`cmd`) и изписвате командата „`python -m http.server [port]`“, където `[port]` е на кой порт да бъде създаден локалния сървър – най-

често се използва 8000 или 8080, но е възможно да се използва и който и да било друг незает порт.



3. Отваряте браузера и въвеждате в търсачката „localhost:[port]“ със съответно избрания порт и отваряте .xml файла.

Резултат:



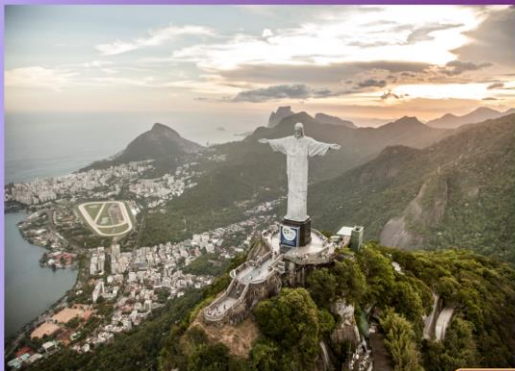
## Каталог на Чудесата на света

Сортиране по структурата на XML

Сортиране по континенти

Сортиране по тип на чудо

Сортиране по азбучен ред



### Име

Статуята на Исус Христос Спасителя

### Дестинация

Южна Америка; Бразилия; Рио де Жанейро; хълм Корковадо

### Работно време

8:00-19:00

### Описание

Статуята на Христос е с разперени ръце и гледа към Рио де Жанейро. Завършена през 1931 г., тя е висока 38 м, изградена е от стоманобетон и е облицована с несмлян талк, устойчив на въздействието на климата. Статуята е разположена на височина 710 м хълм Корковадо в предградието на града. Допълнителната височина на хълма превръща този съвременен колос в емблематична забележителност на града.



### Име

Тадж Махал  
Корона на дворците

### Дестинация

Азия; Индия; Агра;

### Работно време

от 6:00 до 19:00 часа всеки ден, освен петък, когато е затворен за публиката

### Описание

Тадж Махал е мавзолей от бял мрамор на южния бряг на река Ямунa в индийския град Агра. Комплексът включва още джамия и къща за посетители, както и парк, ограден от три страни с масивна стена. Целият комплекс е напълно завършен към 1653 година, а стойността му е около 32 милиона тогавашни рупии (1,4 милиарда лева от 2015 година). Тадж Махал е обявен за Обект на световното наследство на ЮНЕСКО през 1983 година, като „перлата на ислямското изкуство в Индия и един от срещашите всеобщо възхищение шедеври на световното наследство. Смятан е за най-добрият образец на моголската архитектура, съчетаваща ислямски, индийски и персийски елементи и за символ на богатата история на Индия. Главната сграда е симетрична и има висок централен купол, около който са разположени четири по-малки купола. Всеки ъгъл на платформата има по едно минаре, което е леко наклонено навън, за да не падне върху мавзолея при земетресения. Входа към комплекса е през златна арка, която има изписани стихове от Корана. Градната, ляво отсрещна мавзолей, е разположена на четири части, които



Христина Гаджева - ФН7MI0600131

Николай Николаев - ФН8MI0600010



### История на създаването

Историята на Тадж Махал започва през 1631 година, когато Мумтаз Махал умира при раждането на четирнадесетото си дете. Шах Джахан, който е безутешен от загубата на любимата си, решава да построи най-великолепния мавзолей в нейна памет. Той избира мястото на брега на река Ямудна, където той и Мумтаз Махал са прекарвали много щастливи моменти. Наема най-добрите архитектори, художници, занаятчии и майстори от цяла Индия, Персия, Османската империя и Европа, за да работят по проекта. Строителството започва през 1632 година и продължава 22 години, като изисква усилията на около 20 000 души и огромни количества материали и ресурси.

### Промени през годините

През годините Тадж Махал е претърпял различни промени, които са били повлияни от политически, социални и екологични фактори. Ето някои от тях: През 17 век Тадж Махал е бил обект на разбойнически нападения и вандализъм от страна на афгански и персийски войски, които са открадвали и разрушили част от скъпоценните камъни и метали, използвани за украса. През 18 век Тадж Махал е бил занемарен и запустял от слабите и корумпирани моголски императори, които са загубили контрола над Индия. Той е бил използван като квартира за британски войници и чиновици, които са превърнали градината в английски парк и са разрушили някои от съседните сгради. През 19 век Тадж Махал е бил реставриран и консервиран от Британската колониална власт, която е възстановила някои от откраднатите и повредените детайли, е премахнала някои от добавените през 19 век елементи и е възвърнала градината в оригиналния си моголски стил. През 20 и 21 век Тадж Махал е защитаван и поддържан от индийското правителство и различни международни организации, които са се опитали да запазят неговата красота и цялост. Създали са зона за защита на околната среда, ограничили са трафика и туризма в близост до паметника и провеждат редовни почиствания и ремонти.

### Достъпност

За да стигнете до Тадж Махал, имате няколко възможности за транспорт. Една от тях е да пътувате с поезд от Дели до Агра, където се намира паметникът. От станцията Agra Cantonment можете да вземете автобус или таксиметво. Аварийна служба или такси до Тадж Махал.

### Цени на билети

1300 индийски рупии (около 14 евро) за чужденци 50 индийски рупии (около 0.54 евро) за индийски граждани


### Свързани чудеса

- Червената крепост е огромен комплекс от дворци, мечети, салове и музеи, който е построен от моголски император Акбар през 16 век. Тук можете да разгледате различни архитектурни стилове, богата история и култура, както и да се насладите на прекрасна гледка към река Ямудна и Тадж Махал.
- Итимал-уд-Даула е мавзолей, който е известен като "малкият Тадж Махал" или "червеният Тадж Махал". Той е построен от императрица Нур Джахан в памет на баща си Мирза Гияс Бег. Той е един от първите паметници, направени от бял мрамор и украсени с пиетра дура – техника за вграждане на скъпоценни камъни в мрамора.
- Мечетата Джамма е една от най-големите и стари мечети в Индия, която е построена от император Шах Джахан през 1648 година. Тя е изградена от червен пясъчник и бял мрамор и има четири минарета и три купола и може да побере до 25 000 вярващи.

## Каталог на Чудесата на света

Сортиране по структурата на XML
Сортиране по континенти
Сортиране по тип на чудо
Сортиране по азбучен ред

### Природни чудеса:



**Име**  
Гранд Каньон  
Чудото на природата

**Дестинация**  
Северна Америка; САЩ; Аризона; Национален парк Гранд Каньон

**Работно време**  
24 часа в денонощието, цялородишно

**Описание**  
Гранд Каньон е един от най-величествените и впечатляващи природни забележителности на света. Той е дълъг 446 км, широк по 29 км и

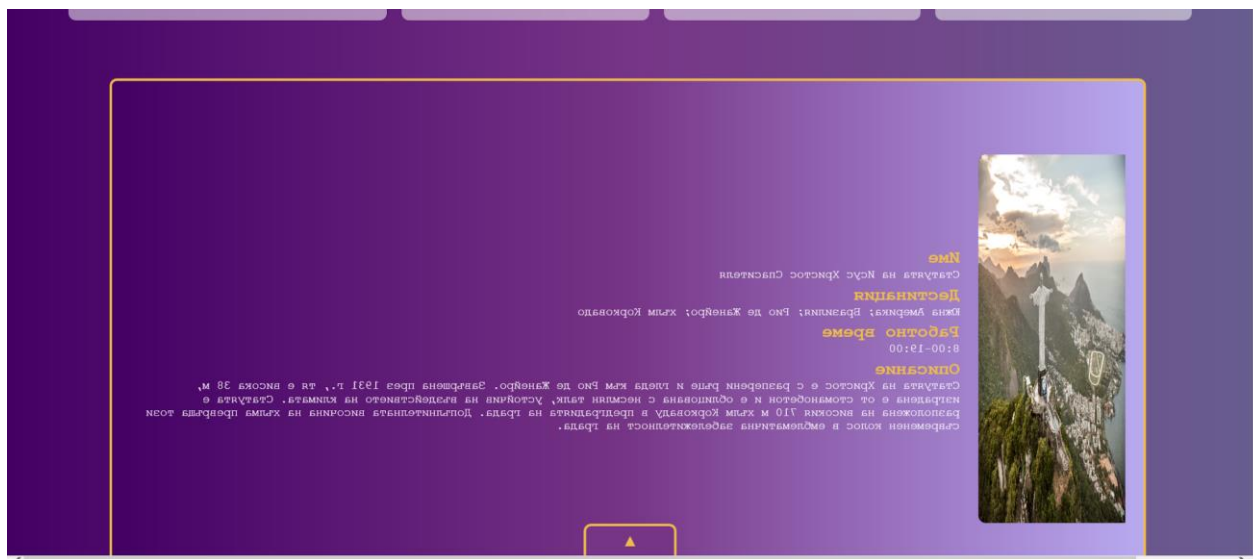
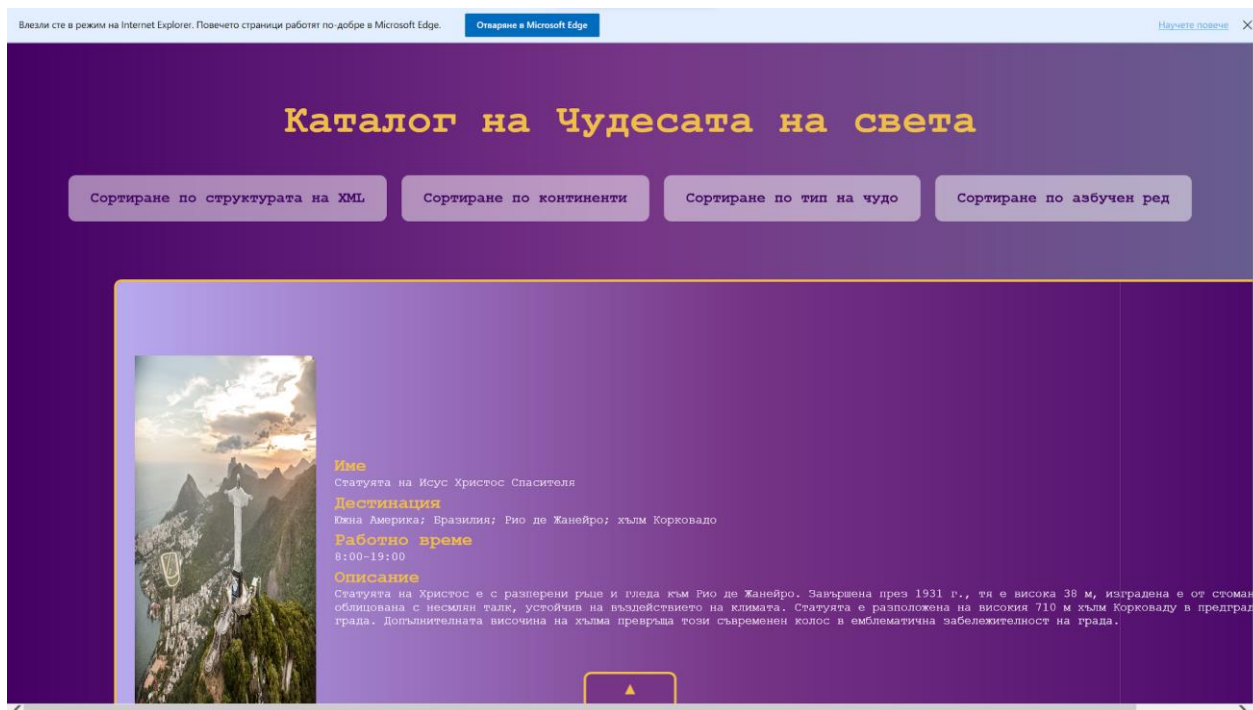
Всички функционалности работят, както са заложиени при проектирането – снимки, бутони, завъртане на картите с информация за чудесата и скролиранията.

## 4.2. Тестване с IE

Това се осъществява чрез приставката на Microsoft Edge, която ни дава възможност да отворим файловете си директно без нуждата от локален сървър. Тук снимките се визуализират, но всички останали компоненти не работят напълно правилно, понеже IE не поддържа някои от новите функционалности на CSS, като въртене на карти, JS функциите и прочие. Приставката е последният символ на изображението.



Резултат:



На втората снимка се вижда липсата на задната част на картата, сортировките с бутоните също не работят. Цялостното оформление също е разтеглено и неправилно.



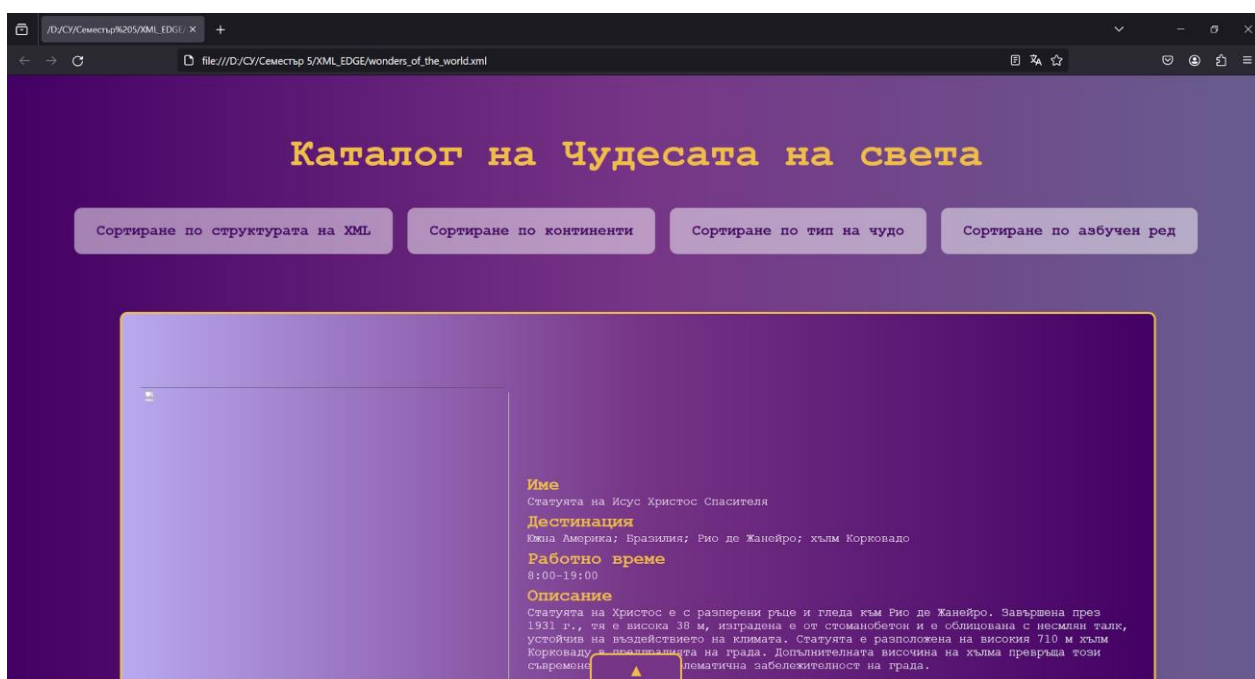
### 4.3. Тестване с Mozilla Firefox

Както при IE, така и в Mozilla Firefox не нужно да се използва локален сървър. Тук обаче се нуждаем от настройките, описани в упражнението за XSLT:

Специално за Mozilla Firefox, извършете следната промяна в настройките:

- отворете локалната страница "about:config"
- натиснете върху бутона за приемане на риска
- потърсете свойството с име `privacy.file_unique_origin`
- ако `privacy.file_unique_origin` е със стойност `true`, променете я на `false`.

Резултат:



Всички функционалности работят както при браузърите Chrome и Edge, но не се визуализират снимките. След направено проучване с документацията на браузъра, установихме, че Mozilla Firefox не поддържа XPath функцията `unparsed-entity-uri()` и снимките не могат да бъдат обработени като entities. Решение само за този браузър е добавянето на следния код в XSL (закоментираната част се поставя в края на `img` елемента). Той обаче не използва entities за представяне на изображенията, поради което считаме, че Mozilla Firefox не удовлетворява напълно изискванията или тоест не преминава през тестването успешно.

```
<!--onerror="this.src='{images/image/image_src/@href}';"-->

```

## 5. Заключение и възможно бъдещо развитие

Проектът е доста обширен и ни предоставя възможност да усвоим знания за уеб технологии и да влезем в дълбини, разширявайки мирогледа си от знания за изготвянето на уеб проект. В процеса на работа се наложи да работим с front-end езици като HTML и CSS, което допринесе за обогатяване на кръгозора ни в областта на програмирането, защото досега сме използвали само backend езици. С изготвянето на курсовата работа упражнихме селектирането и обработката на информация, научихме се как да я структурираме правилно и благодарение на многокомпонентността на проекта, видяхме как данните се обработват от различни езици, минават през различни структури, за да достигнат до финалната фаза на презентиране на уеб сайта.

Използването на езика XML до голяма степен олекоти работата ни, защото първо чрез него форматирахме основната структура на сайта и също така ни позволи да направим валидации на данните чрез включването на DTD и на XML схема. Предимството на XML Schema-та като технология произтича от способността ѝ да поддържа типове данни, които налагат ограничение на съдържанието на някои от елементите. От друга страна, дава възможност за по-голяма гъвкавост и точност за валидирането на данните. Валидацията на DTD чрез използването на entities позволява повторно използване на именувани фрагменти от XML кода, които могат да се извикват на различни места в документа. След валидирането на документа е необходимо използването на XSL за представянето на данните в уеб формат. XSL, съчетан със CSS и HTML, предоставят обширен набор от инструменти за създаване на дизайн и визуализация на проекта в уеб браузър. CSS използва набор от инструменти, чрез които се постига добър външен вид с минимално използване на код. Редакцията и тестването със CSS е бързо и ефикасно, особено при интеграцията му с модерните браузъри.

### 5.1. Възможности за бъдещо развитие

Като за начало, може да се подобри дизайна чрез добавянето на повече бутони или списъци за падащи менюта и да се използват повече анимации, за да се привлече вниманието на клиентите. Друга възможна корекция е по-добро селектиране и представяне на информация, както и добавянето на повече снимков и видео материал за атрактивност.

Идеята за проекта е доста интересна и добра. Бъдещи насоки за развитие могат да бъдат в областта на пътуването и организирането на екскурзия до дестинация, като например се предложат актуални оферти на тур фирми и оператори, които организират почивка до там, или просто да има възможност клиентът да попълни началната точка и датата на тръгване и да се изведе информация за билети за самолетни полети или разписание на автобуси, както и местни хотели със свободни места в този времеви диапазон, където туристите могат да отседнат. По този начин ще улесним още повече тяхното пътуване и ще се даде информация за ценовата категория за почивка до дадена местност. Добра идея ще бъде да се предостави информация за местното население, по-скоро за бита, традициите и обичаите и характерната култура за региона. Също темата може да се разшири към 100 най-известни обекта в света и да се премахне ограничението на топ 7 в различни категории с цел обогатяване на общата култура.

## 6. Разпределение на работата

Екипът, разработил курсовата работа, се състои от двама души - Христина Гаджева 7MI0600131 и Николай Николаев 8MI0600010.

Темата на проекта е избрана според предпочитанията на екипа, така че да бъде интересна и да даде предпоставка за приятен процес на разработка. Процесът по разработването се проведе на няколко етапа като имаше както общи срещи, така и индивидуална работа. Първото събиране послужи за изготвяне на план за работа и определянето на структурата на XML документа. След направата на схематична структура, си разпределихме информацията по равно - за всяка категория се взимаше съответно по 3 или 4 чудеса от човек. После започнахме с намирането, селектирането и обработка на информация. Всеки работеше по своята част. Последва второ събиране, на което сглобихме двата XML документа в един и избрахме подходящи снимки за сайта. Направихме вътрешно DTD и изготвихме XML Schema. По време на третата екипна работа валидирахме XML документа със XML Schema-та в сайта <https://www.xmlvalidation.com/> и оправяхме грешките, които са възникнали. След това Христина Гаджева довърши останалите грешки. Четвъртото събиране беше кратко и се проведе онлайн с помощта на discord -> тук започнахме изработката на xslt и разпределихме задачите до края по следния начин - Николай Николаев да довърши xsl файла и да изработи css, а Христина Гаджева да напише първите точки от документацията. След това се проведе още една среща за направата на сортировките на сайта. Появи се проблем, че сайтът може да се отваря само на FireFox браузър, но не и на Google Chrome и Edge. Николай Николаев се зае със задачата за направата на локален сървър, чрез който да се отвори сайта в горепосочените браузъри. И накрая имаме последна среща за изпипване на дизайна на сайта и разпределяне на задачите по време на представянето - тоест кой за какво ще говори.

Относно документация - разпределихме си точките по следния начин

- Въведение - Христина Гаджева
- Анализ на решението - Христина Гаджева
- Дизайн - Николай Николаев
- Тестване - Николай Николаев
- Заключение и възможно бъдещо развитие - Христина Гаджева
- Разпределение на работата - Христина Гаджева
- Използвани литературни източници и Уеб сайтове - и двамата.
- Апендикс - Христина Гаджева

## 7. Използвани литературни източници и Уеб сайтове

1. Уеб сайт на <https://sites.google.com/site/sedemtechudesas/>
2. Уеб сайт на <https://sedemtechudesas.start.bg/>
3. Уеб сайт на [W3Schools Online Web Tutorials](https://www.w3schools.com/)

4. Множество дискусии от [Stack Overflow - Where Developers Learn, Share, & Build Careers](#) за дребни детайли от различни части на проекта.
5. [Christ the Redeemer | History, Height, & Facts | Britannica](#), [50 Unbelievable Facts About Christ the Redeemer: Ultimate Guide 2024 \(atonce.com\)](#), [18 facts about Christ The Redeemer | FactInformer](#), [A History Of Rio De Janeiro's Christ The Redeemer Monument \(theculturetrip.com\)](#) – за статуята на Христос в Рио де Жанейро
6. [Чичен Ица | Чудесата на света \(chudesatanasveta.com\)](#) [Чичен Ица – туристическият пътеводител Мексико \(dertouristik-reiseleitung.com\)](#) - използван сайт за Чичен ица
7. [Йордания за три дни – Петра | TheTraveler.bg](#) - информация за Петра
8. [Велика китайска стена - Wikiwand](#) - информация за Великата китайска стена
9. [Мачу Пикчу – тайнственият изгубен град на Инките – Чудеса.net \(chudesa.net\)](#) - използвана информация за Мачу Пикчу.
10. [Римският Колизеум: жестокост до съвършенство | Lifebites.bg](#) - информация за Колизеум
11. [Taj Mahal-Official Website of Taj Mahal, Government of Uttar Pradesh \(India\)](#) [Тадж Махал | Определение, история, сайт, история и факти - География И Пътувания | Януари 2024 \(gov-civ-guarda.pt\)](#) [Taj Mahal - UNESCO World Heritage Centre](#) - сайтове за Тадж Махал
12. [Everything to know about Arizona's Grand Canyon National Park \(nationalgeographic.com\)](#), [Grand Canyon National Park | Hiking, Wildlife & Geology | Britannica](#), [How Do I Travel to the South Rim ? - Grand Canyon National Park \(U.S. National Park Service\) \(nps.gov\)](#) – за Гранд Каньон
13. [Aurora \(nationalgeographic.org\)](#), [Aurora Borealis and Aurora Australis | Encyclopedia.com](#), [Auroras Article, Magnetic Storms Information, Auroral Activity Facts -- National Geographic](#) – за Северно и Южно сияние
14. [Mount Everest \(nationalgeographic.org\)](#), [Mount Everest | Height, Location, Map, Facts, Climbers, & Deaths | Britannica](#) – за Еверест
15. [Парикутин - вулканът, изникнал от царевична нива - Lifestyle.bg](#), [Parícutín | Volcano, Mexico, Eruption | Britannica](#), [The Parícutin Volcano, Mexico » Geology Science](#) – за Парикутин
16. [Кои са седемте чудеса на света? | National Geographic България](#), [Рио де Жанейро, Бразилия \(пътеводител за 2024 г.\) – от Travel S Helper](#), [Достопримечателности Рио-де-Жанейро: Топ-20 \(МНОГО ФОТО\) \(tripzaza.com\)](#), – за Пристанището (залива) на Рио де Жанейро
17. [Виктория – грандиозният водопад с кралско име – Пътешественик \(pateshestvenik.com\)](#), [Водопад Виктория в Южной Африке: где находится, фото, описание \(planetofhotels.com\)](#), [Величието на водопада Виктория - На ръба - Peika.bg](#), [Гърмящият дим: Тайните на водопада Виктория | Lifebites.bg](#) – за Водопада Виктория
18. [Great Barrier Reef \(nationalgeographic.org\)](#), [Great Barrier Reef | Map, Animals, Bleaching, & Facts | Britannica](#), [Great Barrier Reef - UNESCO World Heritage Centre](#), [Голям бариерен риф - Wikiwand](#) – Големият Бариерен риф
19. [Разходка из Вавилон - легендарните висящи градини на Семирамида - Spisanie.to](#) [Висящи градини на Вавилон \(jardineriaon.com\)](#) - Висящите градини на Вавилон
20. [ХРАМ НА ЗЕВС ОЛИМПИЙСКИ В АТИНА - ГЪРЦИЯ \(destinationsae.com\)](#) [Вход | Храмът на Зевс Олимпийски | Wonders Of Europe \(wonders-of-europe.com\)](#) - информация за Храм на Зевс в Олимпия

21. [Древният храм на Артемида - история, местоположение и интересни факти - Любопитно \(liubopitnobg.com\)](https://liubopitnobg.com) <https://ekskurzii.biz/efes/efes-vhodna-taksa-cena-rabotno-vreme.html> – храмът на Артемида в Ефес
22. [Мавзолей в Халикарнас - Wikiwand](https://chudesatanasveta.com), [Мавзолеят в Халикарнас | Чудесата на света \(chudesatanasveta.com\)](https://chudesatanasveta.com), [Mausoleum of Halicarnassus | History & Facts | Britannica](https://www.britannica.com/history/article/Mausoleum-of-Halicarnassus), [How the Mausoleum at Halicarnassus became a wonder of the ancient world \(nationalgeographic.com\)](https://www.nationalgeographic.com/history/article/mausoleum-at-halicarnassus-became-a-wonder-of-the-ancient-world) – мавзолей в Халикарнас
23. [Александрийски фар \(280 пр.н.е.\) \(patuvane.info\)](https://patuvane.info), [Lighthouse of Alexandria | History, Location, & Facts | Britannica](https://www.britannica.com/history/article/Lighthouse-of-Alexandria), [Lighthouse of Alexandria - World History Encyclopedia](https://www.worldhistory.org/encyclopedia/Lighthouse_of_Alexandria-World_History_Encyclopedia/), [This Wonder of the Ancient World shone brightly for more than a thousand years \(nationalgeographic.com\)](https://www.nationalgeographic.com/history/article/the-lighthouse-of-alexandria-shone-brightly-for-more-than-a-thousand-years), [The Lighthouse of Alexandria: Why and how was it Built? - World History Edu](https://www.worldhistory.org/encyclopedia/The-Lighthouse-of-Alexandria-Why-and-how-was-it-Built-World_History_Edu/) – Александрийски фар
24. [Colossus of Rhodes - World History Encyclopedia](https://www.worldhistory.org/encyclopedia/Colossus-of-Rhodes-World_History_Encyclopedia/), <https://www.britannica.com/topic/Colossus-of-Rhodes>, [Родоският колос \(istorianasveta.eu\)](https://istorianasveta.eu), [Остров Родос, Гърция. Родоският колос и Акрополът на Линдос \(видео\) — Приключения с Живко Константинов \(jivkokonstantinov.com\)](https://jivkokonstantinov.com) – Родоски колос
25. [Pyramids of Giza | History, Location, Age, Interior, & Facts | Britannica](https://www.britannica.com/history/article/Pyramids-of-Giza), [Великите пирамиди в Гиза | National Geographic България](https://www.nationalgeographic.com/history/article/pyramids-in-giza), [Пирамидите в Гиза Египет | Факти, история, тайни на строежа \(hurghadalovers.com\)](https://www.hurghadalovers.com), [Great Pyramid of Giza - World History Encyclopedia](https://www.worldhistory.org/encyclopedia/Great-Pyramid-of-Giza-World_History_Encyclopedia/) – Пирамидите в Гиза

## 8. Апендикс

За изготвянето на проекта са използвани следните приложения и софтуерни технологии:

- <https://code.visualstudio.com> - среда, на която са писани и съхранявани повечето документи за проекта като xsd, xsl и css файлове.
- Notepad++ - текстов редактор, използван за направата на XML документа
- <https://www.lucidchart.com/pages/> - за изготвянето на диаграми
- <https://www.canva.com/> - за диаграми
- Google Documents - за писането на курсовата документация
- <https://www.python.org/> - използван за локален сървър за отварянето на сайта в Google Chrome и MS Edge
- Mozilla Firefox. Google Chrome. MS Edge - използвани браузъри за визуализиране на проекта