



Софийски университет „Св. Кл. Охридски“

Факултет по математика и информатика

*Бакалавърска програма
„Софтуерно инженерство“*



Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2024/2025 год.

Тема №27: „Каталог на музеите в България“

Курсов проект

Автори:

Камелия Тодорова, фак. номер 3MI0600212

Асен Красимиров, фак. номер 2MI0600231

януари, 2025 г.

София

Съдържание

1	Въведение	3
2	Анализ на решението	4
2.1	Работен процес	4
2.2	Структура на съдържанието	4
2.3	Тип и представяне на съдържанието	6
3	Дизайн	7
4	Тестване	13
5	Заклучение и възможно бъдещо развитие	15
6	Разпределение на работата	16
7	Използвани литературни източници и Уеб сайтове	16
8	Апендикс	17

1 Въведение

Настоящият документ представя проекта „**Каталог на музеи в България**“, разработен в рамките на курса „XML технологии за семантичен Уеб“. Проектът цели да предостави структурирана и леснодостъпна информация за различни музеи в България, използвайки XML и свързаните с него технологии.

1.1 Актуалност на темата

Музеите играят ключова роля в съхраняването и популяризирането на културното наследство. В дигиталната ера е важно тези институции да бъдат представени в удобен за потребителя формат. Създаването на интерактивен каталог на българските музеи чрез XML и XSLT технологии осигурява удобен и стандартизиран начин за достъп до тази информация.

1.2 Решаваният проблем и неговият контекст

Основният проблем, който проектът решава, е липсата на централизирана, добре структурирана и филтрираща информация за музеите в България. В момента потребителите трябва да търсят информация на различни уебсайтове, което е неефективно и често непълно. Нашият проект предоставя унифициран каталог с възможност за филтриране по **регион** и **тип музей**, което улеснява достъпа до релевантна информация.

1.3 Използвана технология

За реализацията на проекта са използвани следните технологии:

- **XML** – за съхранение на информацията за музеите;
- **DTD** – за дефиниране на структурата на XML документа;
- **XSLT** – за трансформиране на XML данните в уебстраница;
- **CSS** – за стилово оформление;
- **JavaScript** – за динамично филтриране на информацията.

1.4 Структура на документа

Останалата част от документа е организирана, както следва:

1. **Описание на XML структурата** – детайлно разглеждане на XML документа и неговите основни елементи.
2. **Определяне на DTD** – анализ на DTD файла, използван за валидация на XML структурата.
3. **XSLT трансформация** – описание на процеса на преобразуване на XML в HTML чрез XSLT.
4. **Стилизиране чрез CSS** – представяне на използваните стилови правила.
5. **JavaScript функционалност** – анализ на динамичните функции за филтриране.
6. **Заклучение** – обобщение на резултатите и възможности за бъдещо разширяване.

2 Анализ на решението

2.1 Работен процес

- Избор на източници и анализ на данните
- Изграждане на структура на съдържанието
- Създаване на правила за валидация
- Изготвяне на съдържанието
- Разработване на стилове за визуализация
- Тестване и оптимизация

2.2 Структура на съдържанието

Проектът „Каталог на музеи в България“ е структуриран по такъв начин, че да осигури ясна категоризация и лесен достъп до информацията. Основната структура на съдържанието е йерархична, като се състои от следните основни елементи:

2.2.1 Основни елементи на структурата

1. Географски региони

- София
- Пловдив
- Варна
- Бургас

2. Типове музеи

- Национални музеи
- Художествени музеи
- Археологически музеи
- Етнографски музеи

3. Музеи (Екземпляри)

- Всеки музей се отнася към определен регион и има тип.

2.2.2 Взаимовръзки в структурата

Връзките между категориите и обектите в XML документа могат да бъдат обобщени чрез следната схема:

- Географски регион (regionId) → съдържа музеи
- Тип музей (typeId) → класифицира музеи
- Музей (museumId) → принадлежи едновременно към регион и тип

2.2.3 Характеристики и атрибути на ресурсите

Географски регион (region)

- Атрибути: regionId – уникален идентификатор на региона
- Честота на срещане: Всеки музей е свързан с точно един регион

Тип музей (type)

- Атрибути: typeId – уникален идентификатор на типа музей
- Честота на срещане: Един музей принадлежи към точно един тип

Музей (museum)

- Атрибути:
 - museumId – уникален идентификатор на музея
 - regionRef – препратка към regionId (връзка към географския регион)
 - typeRef – препратка към typeId (връзка към типа музей)
- Детайли на музея:
 - name – име на музея
 - description – кратко описание
 - images – списък с изображения
 - additionalInfo – контактна информация

2.2.4 Взаимоотношения между елементите

Категория	Връзки с други категории
region	Свързан с множество музеи
type	Свързан с множество музеи
museum	Свързан с един region и един type

2.3 Тип и представяне на съдържанието

Проектът „Каталог на музеи в България“ включва текстово и графично съдържание, като целта е да се предостави богата и визуално атрактивна информация за музеите.

Съдържанието в проекта се състои основно от текстова информация, която описва всеки музей, включително неговото име, кратка характеристика, местоположение, телефон за контакт, работно време и адрес. Тази информация се съхранява в XML формат и се представя чрез XSLT трансформация в уеб страница.

Графичното съдържание включва изображения, свързани с всеки музей. В проекта са използвани **12 графични изображения** във формат **JPEG и PNG**, които визуализират експозициите, архитектурата и интериора на музеите. Изображенията са с резолюция **1200x800 пиксела** и са оптимизирани за уеб представяне.

Файловете са структурирани по следния начин:

- **Текстово съдържание:**
 - XML файл (`museums.xml`), който съдържа информация за музеите
 - DTD файл (`museums.dtd`), дефиниращ структурата на XML документа
 - XSL файл (`museums.xsl`), преобразуващ XML в HTML
- **Графично съдържание:**
 - Изображения (`.jpg` и `.png`), съхранявани в отделна директория
- **Допълнителни файлове:**
 - CSS файл (`style.css`), който определя стилизацията на уеб страницата
 - JavaScript файл (`script.js`), който осигурява динамично филтриране на съдържанието

Всички текстови файлове са кодирани в **UTF-8**, за да поддържат кирилица и специални символи. Графичните файлове са оптимизирани за уеб използване, като размерът на всяко изображение не надвишава 500 KB. В бъдеще може да се добавят мултимедийни елементи като видео или аудио съдържание, ако това е необходимо за разширяване на функционалността на проекта.

3 Дизайн

Проектът „Каталог на музеи в България“ използва **XML технологии** за организиране, представяне и визуализация на информацията за различните музеи. Основната идея е да се създаде **йерархично структуриран XML документ**, който да се валидира чрез **DTD** и да се трансформира в уеб страница чрез **XSLT**.

3.1 Кореновия елемент

Структурата на създадения XML документ следва принципите на добре организиран семантичен уеб ресурс.

Кореновият елемент на XML документа е:

```
<catalogue>
```

Той съдържа два основни поделемента:

- `<regions>` – съдържа списък с географските региони, в които са разположени музеите.
- `<museum-types>` – съдържа списък с типовете музеи.
- `<museums>` – съдържа списък с всички музеи.

3.2 DTD правила

Това е **Document Type Definition (DTD)** файл, който дефинира структурата на XML документа.

- Определя кои елементи са допустими, техните атрибути и подчинени елементи.
- Използва **ID** и **IDREF**, за да валидира връзките между музеи, региони и типове.
- Гарантира, че XML файлът е валиден и спазва зададените правила.

Пример:

- `<catalogue>` трябва да съдържа `<regions>`, `<museum-types>` и `<museums>` в точно тази последователност.

Коренов елемент

```
<!ELEMENT catalogue (regions, museum-types, museums)>
```

Регионите (regions)

```
<!ELEMENT regions (region+)>
<!ELEMENT region (name)>
<!ATTLIST region regionId ID #REQUIRED>
```

- **regionId** – уникален идентификатор (тип ID).
- **name** – съдържа името на региона.

Типове музеи (museum-types)

```
<!ELEMENT museum-types (type+)>
<!ELEMENT type (name)>
```

```
<!ATTLIST type typeId ID #REQUIRED>
```

- **typeId** – уникален идентификатор за типа музей.
- **name** – съдържа името на типа музей.

Музеи (museums)

```
<!ELEMENT museums (museum+)>  
<!ELEMENT museum (name, description, images, additionalInfo)>  
<!ATTLIST museum museumId ID #REQUIRED regionRef IDREF #REQUIRED typeRef IDREF  
#REQUIRED>
```

- **museumId** – уникален идентификатор за музея.
- **regionRef** – препратка към **regionId** (връзка към географския регион).
- **typeRef** – препратка към **typeId** (връзка към типа музей).

3.2 XSD схема

Това е **XML Schema Definition (XSD)** файл, който дефинира структурата и типовете данни в XML документа.

Основни функции:

- **Валидиране на XML документа** на ниво структура и типове данни.
- Задава точни правила за всеки елемент, неговите атрибути и вложени елементи.
- Използва **типове данни** като `xsd:string`, `xsd:ID`, `xsd:IDREF`, за да гарантира коректността на информацията.
- Гарантира, че връзките между музеи, региони и типове музеи са правилно дефинирани.

Кореновият елемент catalogue

```
<xsd:element name="catalogue">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="regions"/>  
      <xsd:element ref="museum-types"/>  
      <xsd:element ref="museums"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Регионите (regions)

```
<xsd:element name="region">
```



```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="regionId" type="xsd:ID" use="required"/>
</xsd:complexType>
</xsd:element>
```

3.3 XML документ

Това е основният **XML файл**, който съдържа данните за всички музеи.

Съдържа йерархично структурирана информация, организирана в три основни секции:

- **<regions>** – Списък с географски региони.
- **<museum-types>** – Типове музеи (национални, художествени и т.н.).
- **<museums>** – Конкретни музеи с детайлна информация.

Включва **атрибути за идентификатори** (ID, IDREF) за свързване между регионите, типовете музеи и отделните музеи.

Осигурява входящите данни за трансформацията и визуализацията.

Пример от кода:

```
<museum museumId="NHM" regionRef="SOF" typeRef="NAT">
  <name>Национален исторически музей</name>
  <description>Музейът представя историята на България.</description>
  <images>
    <image>nhm1.jpg</image>
    <image>nhm2.jpg</image>
  </images>
  <additionalInfo>
    <phone>02 955 4280</phone>
    <workingHours>10:00-18:00</workingHours>
    <address>София, ул. Витошко лале 16</address>
  </additionalInfo>
</museum>
```

3.4 XSL документ

Това е **XSLT (eXtensible Stylesheet Language Transformation)** файл, който преобразува XML документа в HTML формат.

- Създава таблица за представяне на информацията за музеите в браузър.
- Извежда детайлите за всеки музей, включително име, описание, работно време и изображения.
- Осигурява динамично зареждане на изображения чрез тагове.

Примерни функционалности:

- Филтриране по регион и тип музей.
- Поддръжка на изображения, които се показват директно в таблицата.

```
<xsl:template match="museum">
  <tr>
    <td><xsl:value-of select="name"/></td>
    <td><xsl:value-of select="description"/></td>
    <td><xsl:value-of select="additionalInfo/phone"/></td>
    <td><xsl:value-of select="additionalInfo/workingHours"/></td>
    <td><xsl:value-of select="additionalInfo/address"/></td>
    <td>
      <xsl:for-each select="images/image">
        
      </xsl:for-each>
    </td>
  </tr>
</xsl:template>
```

- Прехвърляне на XML данни в HTML таблица.
- Динамично зареждане на изображения.

style.css

Това е **CSS файл**, който дефинира стилизацията на уеб страницата.

- Осигурява визуално оформление на таблицата и текста.
- Включва стилове за:
 - Таблици (граници, цветове, шрифтове).
 - Бутони за филтриране.
 - Ховър ефекти за интерактивност.
- Гарантира адаптивен дизайн, така че страницата да изглежда добре на различни устройства.

script.js

Това е **JavaScript файл**, който добавя интерактивност към уеб страницата.

- Управлява филтрирането по регион и тип музей.
- Скрита/показвани редове в таблицата в зависимост от избрания филтър.
- Повишава потребителското изживяване чрез динамично управление на съдържанието.

images

Това е папка, съдържаща изображения, които се използват за визуализация на музеите.

- Включва файлове в **JPEG** и **PNG** формат.
- Всяко изображение е свързано с музей чрез **XML елемент <image>**.
- Поддържа визуално привлекателно представяне на данните.

3.5 Финален резултат

Финалният резултат от трансформацията е **уеб страница** с каталог на музеите, която включва:

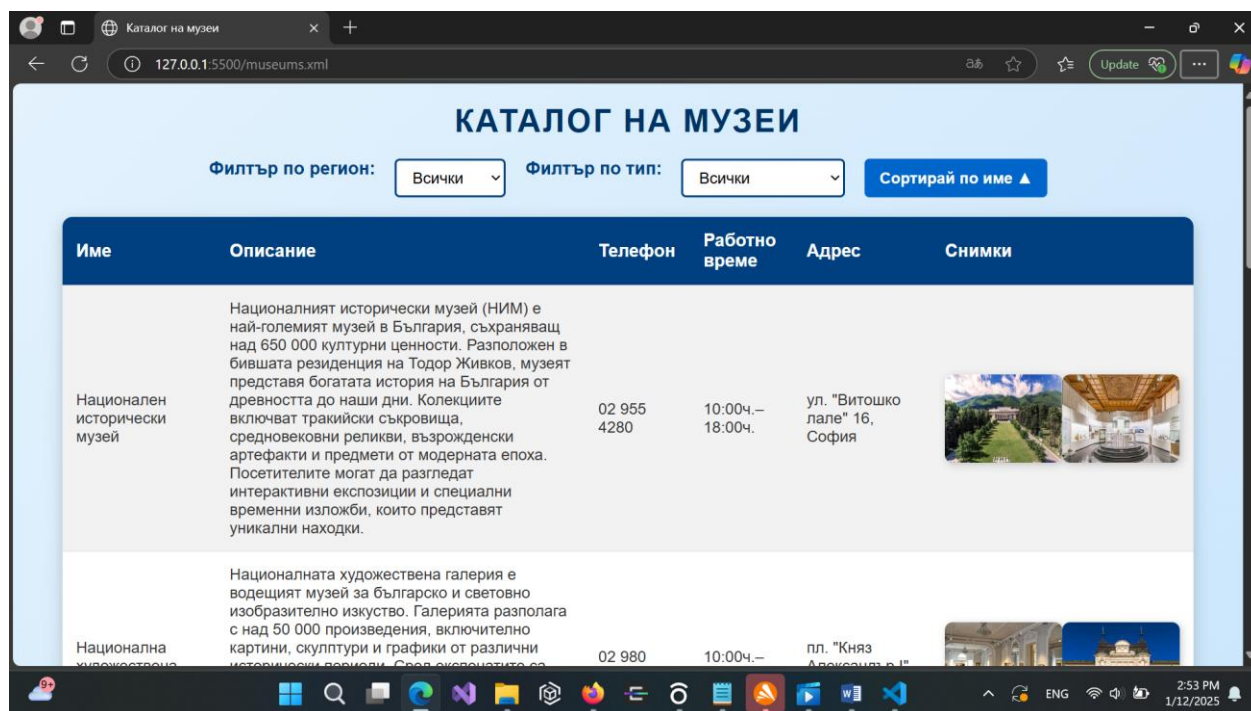
- **Филтриране** по регион и тип музей.
- **Таблично представяне** на информацията.
- **Изображения** за всеки музей.

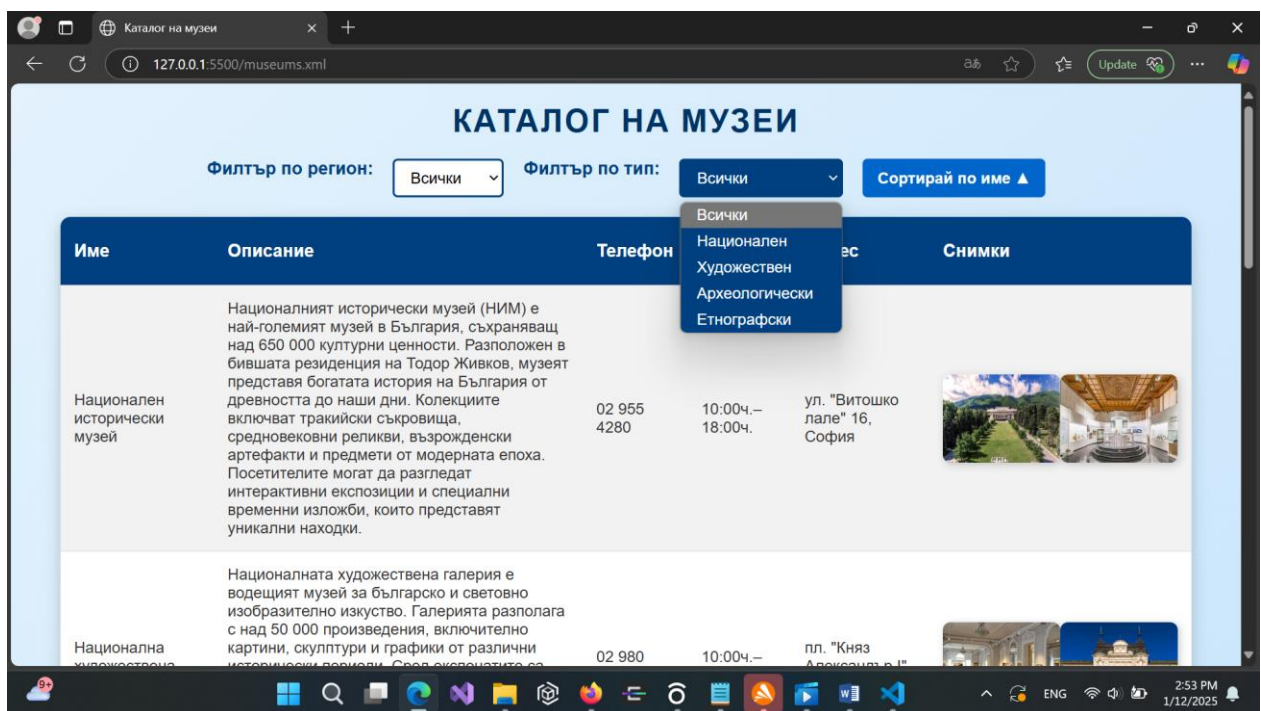
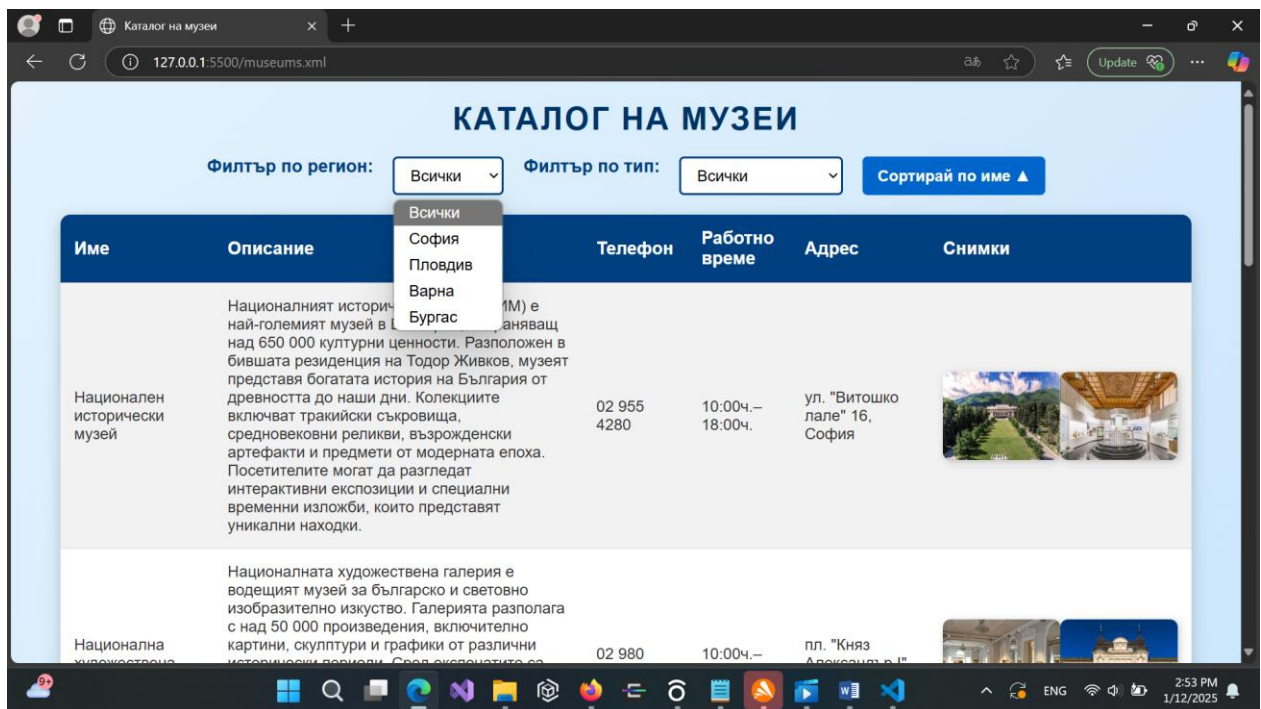
Визуализация в браузър

След стартиране на **локален сървър** и зареждане на **museums.xml**, браузърът показва динамично генерирана уеб страница със стилово оформление, базирано на **CSS**.

Заклучение

Проектът „Каталог на музеи в България“ използва **XML, DTD, XSLT и JavaScript** за създаване на динамичен и интерактивен уеб ресурс. Чрез валидация на **XML документа**, коректно прилагане на **XSL трансформации** и **CSS стилове**, постигаме лесна навигация и визуално привлекателно представяне на съдържанието.





4. Тестване

Изготвеното решение беше многократно тествано по време на разработката. Бяха създадени тестови **XML документи**, които потвърдиха правилното валидиране на **DTD** схемата. След изграждането на **XSL трансформацията** бяха извършени ръчни тестове за правилната визуализация на данните в различни браузъри.

Тестване на различни сценарии

Тестването беше извършено при два различни сценария:

1. **Директно отваряне на XML файла от локалната файлова система** – Някои браузъри не поддържат директно прилагане на XSLT към XML файлове.
2. **Зареждане на XML файла чрез локален уеб сървър** – Позволява правилна обработка и визуализация на XSLT трансформацията.

Тествани браузъри и резултати

Google Chrome

✓ Резултат:

- Не визуализира XSLT директно от локалната файлова система.
- Работи коректно при зареждане през локален уеб сървър (<http://localhost/>).
- Без бъгове или визуални несъответствия.

Mozilla Firefox

✓ Резултат:

- Не работи при директно отваряне на XML файла.
- Може да бъде зареден само чрез локален уеб сървър (<http://localhost/>).
- Без открити бъгове при сървърно зареждане.

Microsoft Edge

✓ Резултат:

- Не визуализира XSLT при нормален режим.
- Работи коректно при зареждане през локален уеб сървър.
- Няма забележки при съвместимост с модерни CSS стилове.

Opera

✓ Резултат:

- Не визуализира XSLT директно от локалната файлова система.
- Работи коректно при зареждане през уеб сървър.
- Без открити бъгове или несъответствия.

Заклучения от тестването

Проектът работи **правилно в Chrome, Firefox, Edge и Opera**, но изисква зареждане чрез **локален уеб сървър** - <http://localhost/>. Това гарантира коректното изпълнение на XSL трансформацията и визуализацията на съдържанието.

Валидация на XML документа

За валидация на изготвения **XML документ** спрямо създадената **DTD** схема беше използван **онлайн валидатор** <https://xmlvalidation.com>.

Стъпки за валидация:

1. Качваме **XML файла** (`museums.xml`).
2. Избираме отметката **Validate against external XML schema**.
3. Качваме **DTD файла** (`museums.dtd`).
4. Стартираме валидацията.

✓ **Резултат:** XML документът премина успешно всички проверки, като **не бяха открити грешки** в структурата и данните.

Тези тестове потвърдиха коректната работа на проекта и неговата съвместимост с XML стандартите и уеб браузърите.

5. Заключение и възможно бъдещо развитие

Проектът „Каталог на музеи в България“ демонстрира как XML технологиите могат да бъдат използвани за създаване на добре организирана, достъпна и интерактивна уеб-базирана система за представяне на информация. Чрез използването на **XML, XSLT, CSS и JavaScript** се постига структурирано и визуално атрактивно съдържание, което може лесно да бъде разширявано и адаптирано според нуждите на потребителите.

Основни резултати от проекта

1. **Структуриран XML каталог** – Създадена е XML база данни, която съдържа информация за различни музеи, включително тяхното описание, местоположение, тип и изображения.
2. **Динамично филтриране на съдържанието** – Чрез JavaScript и XSLT потребителите могат лесно да търсят и филтрират музеите по регион и тип.
3. **Оптимизирана визуализация** – Чрез CSS дизайнът на страницата е адаптиран за по-лесно възприемане на информацията.

4. **Лесна интеграция и разширяемост** – Структурата на проекта позволява добавяне на нови музеи без промяна на основния код.

Възможности за бъдещо развитие

Въпреки че проектът предлага солидна основа за представяне на музеи, той може да бъде разширен в няколко насоки:

- **Добавяне на допълнителни категории и метаданни** – Включване на информация за билети, рейтинги, потребителски рецензии и интерактивни турове.
- **Интерактивна карта** – Внедряване на географска карта с местоположенията на музеите за по-интуитивно представяне.
- **Поддръжка на различни езици** – Добавяне на многоезична поддръжка, така че потребителите да могат да избират език за визуализация.
- **API интеграция** – Възможност за свързване с външни системи за културно наследство, като Open Data платформи за културен туризъм.
- **Мобилна версия** – Разработване на адаптивен дизайн или мобилно приложение за по-добро потребителско изживяване на мобилни устройства.
- **Добавяне на мултимедийно съдържание** – Включване на видео клипове, аудио гидове и виртуални турове за по-дълбоко преживяване.

Заклучение

Проектът „Каталог на музеи в България“ предоставя функционален и ефективен начин за организиране и визуализиране на информация за музеи чрез **XML технологии**. Той не само улеснява достъпа до важна културна информация, но и предлага потенциал за бъдещо разширяване и подобрене чрез допълнителни интерактивни функции. В бъдеще този проект може да бъде адаптиран и за други културни институции, като галерии, археологически обекти и исторически забележителности, което ще допринесе за по-широко популяризиране на българското културно наследство.

6. Разпределение на работата

По време на работата по проекта си разпределихме задачите така, че и двамата да имаме равен принос, като някои части разработихме заедно, а други си разделихме. Целта беше да комбинираме усилията си, за да получим добре структуриран и работещ проект.

- Създаване на структура на документа - съвместно
- XML файлът – работихме заедно по създаването на структурата на XML документа
- XSLT трансформацията – разработихме я съвместно, като си помагахме за стилизирането и логиката на преобразуването от XML към HTML
- XSD и DTD валидацията – съвместно
- JavaScript функционалността – Асен Красимиров
- Тестване и финални подобрения- съвместно

- Документация - съвместно

7. Използвани литературни източници и уеб сайтове

1. Официален уебсайт на Националния исторически музей – <https://www.historymuseum.org>
2. Официален уебсайт на Националната художествена галерия – <https://www.nationalgallery.bg>
3. Официален уебсайт на Археологическия музей в Пловдив – <https://www.archaeologicalmuseumplovdiv.org>
4. Официален уебсайт на Варненския археологически музей – <https://www.archaeo.museumvarna.com>
5. Официален уебсайт на Регионалния исторически музей – Бургас – <https://www.burgasmuseums.bg>

Литературни източници:

6. Петров, Г. (2018). *История на българските музеи*. София: Университетско издателство
7. Иванова, М. (2020). *Културното наследство на България*. Пловдив: Академично издателство
8. Димитров, К. (2019). *Съвременни подходи за дигитализация на културното наследство*. Варна: Издателство „Наука“

8. Апендикс

Инструкции за стартиране на проекта

Изисквания:

Необходим е локален уеб сървър за правилното зареждане на проекта. Поддържаните браузъри са **Firefox, Chrome, Edge и Opera** (с локален хост).

Стартиране чрез Python:

1. Отваряме терминал или **Command Prompt**.
2. Отиваме до папката, в която се намират файловете (**museums.xml, museums.xsl, museums.dtd**).

```
cd "път/до/папката"
```

3. Стартираме локален сървър:

- Ако има **Python 3**:

```
python -m http.server 8000
```

- Ако има **Python 2**:


```
python -m SimpleHTTPServer 8000
```

4. Отваряме **Google Chrome, Firefox или Opera** и в адресната лента въвеждаме:

```
http://localhost:8000/museums.xml
```

Сега проектът ще се зареди правилно и **XSLT** ще се приложи.

Описание на структурата на проекта

Използвани технологии и библиотеки