



Софийски университет „Св. Кл. Охридски“

Факултет по математика и информатика

*Бакалавърска програма
„Софтуерно инженерство“*



Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2025/2026 год.

Тема №39: „Каталог на видеоигри за развлечение - 2“

Курсов проект

Автори:

Ивайло Кънчев, фак. номер 2MI0600305

Йоан Байчев, фак. номер 0MI0600328

януари, 2026 г.

София

Съдържание

1	Въведение	1
2	Анализ на решението	2
2.1	Работен процес	2
2.2	Структура на съдържанието	3
2.3	Диаграма на модела (ER)	5
2.4	Тип и представяне на съдържанието	6
3	Дизайн	6
3.1	Архитектура и компоненти	6
3.2	XML ентитети	7
3.3	Ключове и референции	7
3.4	Валидация и тестове	7
3.5	Поток и роля на компонентите	8
3.6	Диаграма на валидация и рендер	8
3.7	Негативен сценарий при валидация	9
3.8	Диаграма на потока	10
3.9	Разширяемост и алтернативи	10
3.10	Рискове и ограничения	10
3.11	Обобщение	11
4	Използване на изкуствен интелект	13
5	Тестване	13
5.1	Валидация на XML съдържанието	13
5.2	Тестване на XSLT трансформацията	14
5.3	Тестване на визуализацията и стилизирането	14
5.4	Тестване на интерактивната функционалност	14
5.5	Обобщение на резултатите	14
6	Заклучение и възможно бъдещо развитие	17
7	Разпределение на работата	18

8	Източници и литература	18
9	Апендикс	19

1. Въведение

Настоящият документ представя разработката на курсов проект по „XML технологии за семантичен Уеб“, чиято цел е създаване на каталог на видеоигри за развлечение. Проектът показва как чрез XML и свързаните с него технологии (XML Schema, XSLT, CSS и JavaScript) може да се моделира, валидира и визуализира богато текстово и графично съдържание за популярни игри – жанрове, компании (разработчици и издатели), платформи, дати на излизане и изображения. Основният проблем, който решаваме, е структурираното и консистентно представяне на многоизмерни данни за видеоигри, така че те да могат да се валидират автоматично и да се визуализират интерактивно без сървърна логика.

В контекста на предмета това демонстрира пълния технологичен цикъл: дефиниране на схема с ключове и референции, използване на непарснати XML ентитети за изображения, трансформация на съдържанието с XSLT директно в браузър и добавяне на динамика с JavaScript за търсене, филтриране и сортиране. Решението използва единичен XML документ с три основни колекции (жанрове, компании, игри), който се валидира срещу XML Schema с ключове и keyref зависимости (ID/IDREF и композитен ключ за компаниите). Изображенията са свързани чрез непарснати ентитети, за да се осигури ясна връзка между данните и графичните ресурси. Визуализацията е изцяло клиентска: XSLT оформя HTML таблица, CSS придава стил, а JavaScript реализира интерактивните функции.

Проектът е актуален като учебен пример за XML-first подход: показва цялата верига от модел и схема, през валидация, до клиентска трансформация без бекенд. Проблемът, който се адресира, е нуждата от консистентно и валидируемо представяне на богати данни за видеоигри с възможност за офлайн визуализация и интерактивност директно в браузъра.

Останалата част от документа е структурирана така: раздел 2 описва работния процес и структурата на съдържанието; раздел 3 разглежда дизайна и архитектурата на решението; раздел 4 описва използването на изкуствен интелект в процеса; раздел 5 представя тестването и резултатите; раздел 6 обобщава заключенията и очертава възможности за бъдещо развитие; раздели 7–9 съдържат разпределението на работата, източниците и апендикси.

Цели и критерии за успех:

Цел	Реализация
Стриктна валидация	<code>xmllint -schema catalog.xsd catalog.xml</code> , ключове/референции, <code>enum</code> за ENTITY
Клиентска интерактивност	XSLT генерира HTML + JS (търсене, филтър, сортиране)
Графика през ENTITY	Непарснати PNG ентитети + <code>unparsed-entity-uri()</code>
Офлайн работа	Един XML + XSD + XSLT + CSS + PNG, без бекенд; може през локален сървър
Разширяемост	Добавяне на игра/изображение с обновяване на DOCTYPE/XSD <code>enum</code> , без промяна в XSLT/JS

2. Анализ на решението

2.1 Работен процес

Работният процес на „Каталог на видеоигри за развлечение“ обхваща целия цикъл от подготовката на данните, през валидацията и трансформацията, до интерактивната визуализация в браузър. Входният документ `catalog.xml` съдържа три колекции: жанрове (`genres`), компании (`companies`) и игри (`games`). Всеки жанр има уникален идентификатор и име; всяка компания — име, държава и роля; всяка игра реферира към жанр, към една или повече компании по роля, съдържа заглавие, платформи, резюме, дата на излизане и препратка към изображение чрез XML ентитет. Ентитетите за изображения (PNG файлове в директорията `images/`) са дефинирани в DOCTYPE като непарснати, за да се поддържа ясно разграничение между данни и медийни ресурси.

Този процес може да се разглежда като тройка „вход–обработка–изход“. **Входът** е един XML документ с данни + DOCTYPE ентитети + XSD. **Обработката** включва (1) валидация срещу XSD; (2) XSLT трансформация в браузър; (3) зареждане на CSS и инициализация на JavaScript за интерактивност. **Изходът** е HTML страница с таблица и контроли за търсене/филтър/сортиране, която може да се отвори директно или през прост локален сървър. Тъй като няма бекенд, целият жизнен цикъл е файлов и преносим.

Първата стъпка е валидиране срещу XML Schema (`catalog.xsd`), която дефинира типове за идентификатори, ключове и референции, както и изброени стойности за атрибута `image`. Ключовете `genreKey` и `companyKey` (композиционен ключ по `name+role`) и съответните ключови връзки (`keyref`) гарантират, че всяка игра сочи валиден жанр и валидни компании за разработка/издаване. След валидирането XML се трансформира чрез XSLT стил (`catalog.xsl`) директно в браузър. Трансформацията създава HTML таблица с колони като заглавие, дата на излизане, жанр, платформи, разработчик, издател, описание и изображение; изображенията се вмъкват чрез функцията `unparsed-entity-uri(@image)`. JavaScript осигурява интерактивност: търсене, филтър и сортиране. CSS определя визуалния облик и удобството за ползване.

Контрол на грешки: ако `xmllint` засече липсващ жанр или компания, процесът спира и XML се коригира преди трансформация. Ако в DOCTYPE липсва ентитет, XSD енума отхвърля стойността и улавя несъответствието. При визуализация, счупен `src` би се видял като празно изображение, което се предотвратява чрез предварителната валидация.

Изходът е една HTML страница, която може да се зареди директно в браузър без сървърна логика. Всички зависимости са файлове: XML, XSD, XSLT, CSS и изображения. Работният процес е лесно повторим: при добавяне на нова игра се обновява XML-ът и, ако има ново изображение, се актуализира DOCTYPE/XSD енума; валидирането и трансформацията се извършват отново без промени в кода.

Примерен поток (конкретно): входът е `catalog.xml` + `catalog.xsd` + PNG в `images/`. Стъпки: (1) `xmllint` валидира; (2) браузър прилага `catalog.xsl`; (3) генерира се HTML с таблица и контроли; (4) JS активира търсене/филтри/сортиране; (5) потребителят преглежда/филтрира списъка. Очакван резултат: осем реда, всеки

със снимка, коректни жанрове и компании.

Предимства на този поток: няма зависимости от сървър, така че каталогът се пренася и отваря офлайн; валидирането се случва преди рендериране, което гарантира качеството на данните; обновяването е локализирано в един XML файл плюс, при нужда, DOCTYPE/XSD. Ограничения: при добавяне на много ресурси размерът на XML и изображенията може да забави зареждането; филтър/търсене работят в брауъра и не е оптимизирано за хиляди записи (но е достатъчно за учебния обем).

Мини-сценарии (оперативни):

- Добавяне на нова игра: PNG → ENTITY в DOCTYPE → нова стойност в XSD enum → елемент `game` с валидни `genreId/companyRef` → `xmllint`.
- Актуализация на изображение: заменя се PNG + се променя SYSTEM пътят в DOCTYPE; XSD enum остава същия, ако името на ENTITY не се сменя.
- Промяна на компании: добавяне на `company (name+role)` преди да се реферира в `companyRef`; XSD блокира липсващи записи.

2.2 Структура на съдържанието

Структурата на съдържанието е описана формално в `catalog.xsd`. Моделът включва:

- **Жанрове** `genres/genre` – атрибути `id` (тип ID) и `name`. Жанрвете са седем: Action-adventure, Action RPG, First-person action, Sandbox/Survival, Puzzle, Platformer, Role-playing. Ключ `genreKey` гарантира уникалност на идентификаторите, а игрите реферират към жанровете чрез `genreId`.
- **Компании** `companies/company` – атрибути `name`, `country` и `role`. Композитният ключ `companyKey (name+role)` позволява една компания да бъде едновременно разработчик и издател. В игрите елементът `companyRef` съдържа име и роля и се валидира чрез ключова връзка `companyRefKey`.
- **Игри** `games/game` – атрибути `id`, `genreId`, `image` (ENTITY, ограничен чрез `ImageEntityType`) и `releaseDate`. Поделемента `title`, `platforms`, `summary` и `companies/companyRef`. Структурата допуска множество разработчици и издатели, а датите са в ISO формат (YYYY-MM-DD) за улеснено сортиране.
- **Медийни ресурси** – в DOCTYPE са дефинирани непарснати ентитети (`totk`, `eldenring`, `doometernal`, `minecraft`, `tetris`, `witcher3`, `mario`, `bg3`), които сочат към PNG файлове. XSD енума `ImageEntityType` дублира тези стойности и гарантира съответствие.

Връзките между колекциите се реализират чрез ID/IDREF и `key/keyref`, което позволява автоматична проверка за консистентност. Форматът е лесно разширяем: добавяне на нови игри, жанрове или компании изисква само попълване на XML-а и, при ново изображение, обновяване на DOCTYPE/XSD.

Проверки за консистентност: дублиране на жанр или компания по роля се блокира от ключовете; липсващ `genreId` се улавя от `gameGenreRef`; грешен `image` се улавя от `ImageEntityType`; неправилно въведена дата (не-ISO) се отхвърля от `xs:date`. Така

всяка връзка има формална защита, а данните остават „самодокументирани“ чрез схемата.

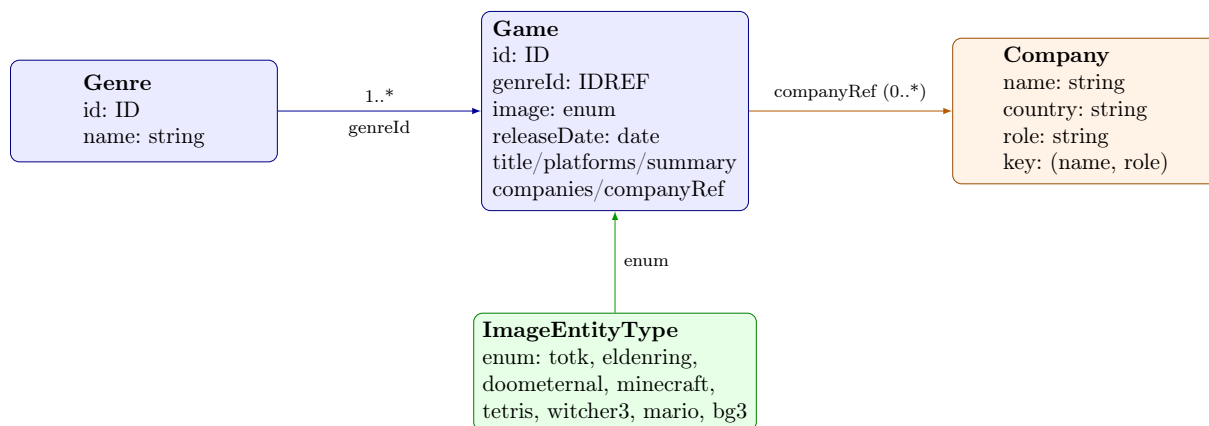
Примери за валидни/невалидни данни и очаквана реакция:

Поле	Невалидно	Резултат от XSD/валидатор
game/@genreId	Стойност, която не съществува в genres/genre/@id	keyref gameGenreRef: Element 'game', attribute 'genreId' нарушава ключ
companyRef/@name+@role	Компания липсва или роля не съвпада	keyref companyRefKey нарушен
game/@image	Стойност извън енума (напр. halo)	cvc-enumeration-valid грешка
game/@releaseDate	Неформатна дата (напр. 2023/12/01)	cvc-datatype-valid.1.2.1: value is not facet-valid with respect to pattern
genres/genre/@id	Повтарящ се ID (дубликат)	Duplicate unique value за genreKey

Обзор на игрите (кратко):

- game1 – The Legend of Zelda: TOTK, жанр Action-adventure, разработчици Nintendo EPD/Monolith Soft, издател Nintendo, дата 2023-05-12.
- game2 – Elden Ring, жанр Action RPG, FromSoftware (dev/publisher), Bandai Namco (publisher), дата 2022-02-25.
- game3 – DOOM Eternal, жанр First-person action, id Software / Bethesda, дата 2020-03-20.
- game4 – Minecraft, жанр Sandbox/Survival, Mojang (dev/pub), дата 2011-11-18.
- game5 – Tetris, жанр Puzzle, Alexey Pajitnov / Vadim Gerasimov / The Tetris Company, дата 1984-06-06.
- game6 – The Witcher 3, жанр Action RPG, CD Projekt RED / CD Projekt, дата 2015-05-19.
- game7 – Super Mario Odyssey, жанр Platformer, Nintendo EPD / 1-Up Studio / Nintendo, дата 2017-10-27.
- game8 – Baldur's Gate 3, жанр Role-playing, Larian Studios (dev/pub), дата 2023-08-03.

2.3 Диаграма на модела (ER)



Фигура 1: ER-скица: жанр → игри (1..*), игри → компании чрез `companyRef`, изображение като енумирана стойност.

Примерен фрагмент (game):

```

114 <game id="game3" genreId="g3" image="doometernal" releaseDate="2020-03-20">
115   <title>DOOM Eternal</title>
116   <platforms>Windows, PlayStation 4, Xbox One, Stadia, Nintendo Switch, PlayStation 5, Xbox Series X</platforms>
117   <summary>
118     A fast-paced first-person shooter and direct sequel to DOOM (2016).
119     Developed by id Software and published by Bethesda Softworks, it
120     launched on 20 March 2020 for Windows, Xbox One, PlayStation 4 and
121     Stadia, later arriving on Nintendo Switch in December 2020 and on
122     next-generation consoles in 2021.
123   </summary>
124   <companies>
125     <companyRef name="id Software" role="developer"/>
126     <companyRef name="Bethesda Softworks" role="publisher"/>
127   </companies>
128 </game>
  
```

Фигура 2: XML фрагмент за `game3` (DOOM Eternal) с атрибути и компании.

Процедура при добавяне на нова игра (поддръжка):

1. Добави PNG в `images/` и дефинирай непарснат ентитет в DOCTYPE.
2. Обнови XSD `ImageEntityType` (enumeration) с новото име.
3. Вмъкни запис в `games` с валиден `genreId` и `companyRef` стойности по роля.
4. По желание добави нов жанр/компания в съответната колекция.
5. Пусни `xmllint` за валидация и визуална проверка в браузър.

Разширяемост: нова игра изисква добавяне на елемент в `games`, валиден `genreId` към `genres`, референции към компании и, при ново изображение, обновяване на DOCTYPE и XSD `enum`. Ако се въведат поджанрове, може да се добави `parentId` и нови `key/keyref`. За платформи може да се дефинира отделна колекция с `enum`, за да се избегнат свободни текстове.

Вид на крайния резултат: едностраничен HTML с таблица на игрите, интерактивни контроли (търсене, филтър по жанр, сортиране), зареждан през локален сървър, с изображения от ENTITY.

2.4 Тип и представяне на съдържанието

Съдържанието комбинира текст и графика. Всички текстови елементи (заглавия, резюмета, списъци с платформи, имена на компании, дати) са в UTF-8 и следват ISO форматите. Резюметата са чист текст без HTML тагове, за да се предотврати инжектиране на код. Графичните ресурси са осем PNG изображения с размер 1.5–3.2 MB, достъпни чрез непарснати ентитети; XSLT ги вмъква с `unparsed-entity-uri()`.

Фронтендът представлява HTML таблица с удобни контроли за търсене и филтриране. CSS добавя стил – светъл фон, цветен хедър, редуващи се редове, ограничена ширина на изображенията. JavaScript осигурява динамика без нужда от презареждане.

Файлове и роли:

Файл	Роля
catalog.xml	Данни: жанрове, компании, игри, връзки към изображения (ENTITY)
catalog.xsd	Схема: типове, ключове, keyref, enum за изображения
catalog.xsl	Трансформация към HTML, вграден JS за сортиране/филтър/търсене
style.css	Стилове за таблицата, контролите и изображенията

3. Дизайн

3.1 Архитектура и компоненти

Проектът е реализиран като изцяло клиентско приложение, което не изисква сървър или база данни. Основните компоненти са:

1. **Данни** – `catalog.xml` съдържа всички записи за жанрове, компании и игри. DOCTYPE определя връзките към изображенията.
2. **Схема** – `catalog.xsd` дефинира типове, ключове и ключови връзки, като гарантира валидност на данните.
3. **Трансформация** – `catalog.xsl` генерира HTML и включва JavaScript за интерактивност; `style.css` задава стил.
4. **Резултат** – една HTML страница, която може да се отвори директно и включва търсачка, филтър по жанр и сортиране на колони.

Данните и изгледът са отделени: XML/XSD описват модела, XSLT е „view“ слой, а CSS/JS оформят и добавят динамика. Пътят на потока е еднопосочен (XML → XSLT

→ HTML/JS/CSS), което улеснява поддръжката. Всяка промяна в данните не налага промяна в кода, освен ако не се въвеждат нови типове ресурси (изображения, нови полета).

3.2 XML ентитети

Медийните ресурси са дефинирани като непарснати ентитети в DOCTYPE:

```
<!ENTITY totk          SYSTEM "images/totk.png"          NDATA png>
<!ENTITY eldenring     SYSTEM "images/eldenring.png"     NDATA png>
...
```

Използването на ентитети позволява централизирано управление на изображенията. В XSLT се използва `unparsed-entity-uri(@image)` за превръщане на ентитета в URL за ``, което улеснява поддръжката и валидирането чрез XSD енума `ImageEntityType`.

3.3 Ключове и референции

Име	Тип	Обхват / Значение
genreKey	xs:key genres/genre/@id върху	Гарантира уникални жанрове; <code>game/@genreId</code> сочи към него.
gameGenreRef	xs:keyref game/@genreId върху	Всяка игра реферира валиден жанр.
companyKey	xs:key върху company/@name + @role	Композитен ключ (фирма, роля), позволява една фирма да е dev/publisher.
companyRefKey	xs:keyref върху companyRef/@name + @role	Проверява, че всеки <code>companyRef</code> сочи към валидна компания/роля.
ImageEntityType	xs:simpleType enumeration с	Синхронизира имена на ENTITY от DOCTYPE с XSD, предотвратява счупени изображения.

3.4 Валидация и тестове

Преди трансформацията данните се валидират с помощта на команда:

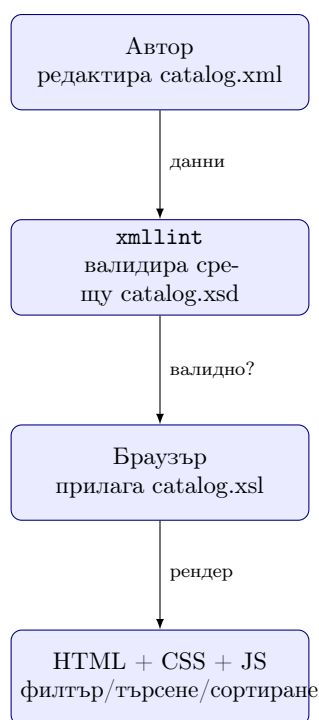
```
xmllint --noout --schema catalog.xsd catalog.xml
```

Това потвърждава правилно изпълнени ID/IDREF връзки, композитни ключове и типове. След трансформацията се извършва визуална проверка в няколко браузъра (Chrome, Firefox, Edge), за да се гарантира коректно зареждане на изображенията и работещи функции за търсене, филтър и сортиране.

3.5 Поток и роля на компонентите

1. Зареждане на `catalog.xml` в браузър, при което се прилага `catalog.xsl`.
2. XSLT генерира HTML таблица и вгражда JavaScript; `unparsed-entity-uri()` резолвира `@image` към PNG.
3. CSS придава стил; JS активира филтър по жанр, търсене и сортиране (стрингово и по дати).
4. При промяна на данни: обновяване на XML (+ DOCTYPE/XSD enum при ново изображение) и повторно валидиране; трансформацията остава непроменена.

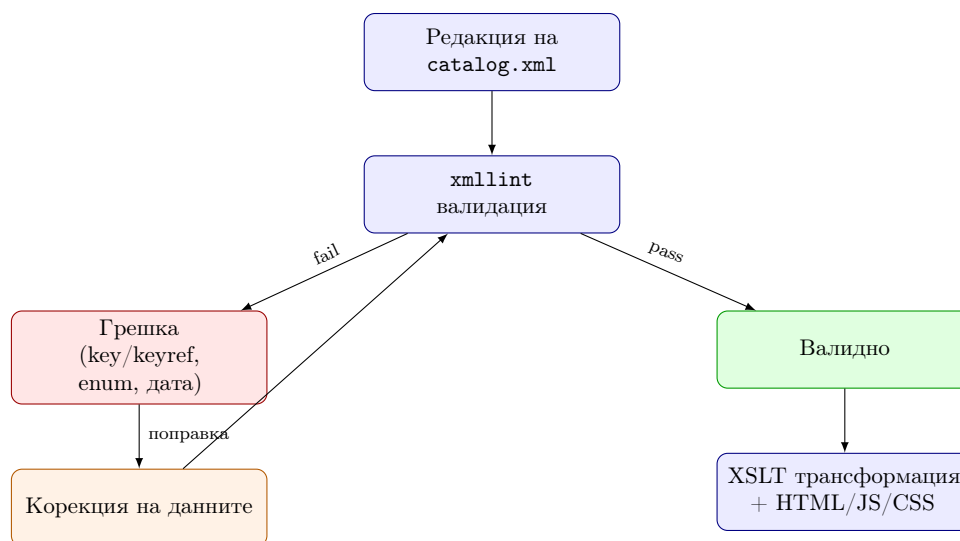
3.6 Диаграма на валидация и рендер



Фигура 3: Поток при актуализация: редакция → валидация с `xmllint` → XSLT трансформация в браузър → интерактивен интерфейс.

Диаграмата показва типичния успешен път: редакция на данните, валидация срещу схемата и автоматична трансформация в браузъра без допълнителен бекенд.

3.7 Негативен сценарий при валидация



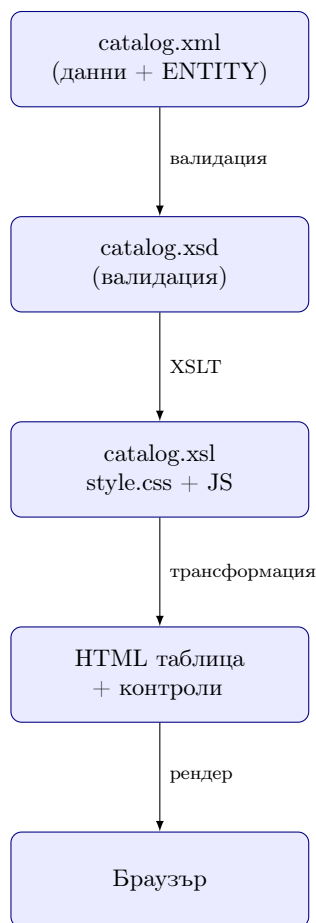
Фигура 4: При неуспех на XSD валидацията процесът спира, коригира се XML-ът и се валидира отново.

При грешка валидирането прекъсва потока; корекцията връща документа към валидно състояние преди трансформацията.

Чести причини за провал и корекция:

- Пропуснат `genreId` или грешно изписан идентификатор → добавяне/поправка в `genres` и обновяване на атрибута в `game`.
- `companyRef` към несъществуваща комбинация `name+role` → добавяне на `company` или поправка на ролята (`developer` / `publisher`).
- Ново изображение без актуализиран `ImageEntityType` или `DOCTYPE ENTITY` → добавяне на `enumeration` и `ENTITY` декларация.
- Невалидна дата (формат `YYYY-MM-DD`) → поправка към `ISO` формат.

3.8 Диаграма на потока



Фигура 5: Еднопосочен поток: данни → валидация → трансформация → визуализация в браузър.

Потокът е еднопосочен и файлов: от XML с ENTITY към XSD валидация, после XSLT рендер и клиентска интерактивност.

3.9 Разширяемост и алтернативи

Разширения: поджанрове и PEGI/ESRB рейтинги; филтри по платформи и диапазон дати; PDF изход чрез XSL-FO; локализация на интерфейса; по-строга типизация на платформи чрез отделен `enum`/колекция; интеграция с външни API. **Алтернативи:** JSON/HTML без XSLT (по-малко демонстрация на XML трансформации); бекенд/БД (по-тежка инфраструктура); външни шрифтове/фреймуърци (добавят зависимости).

3.10 Рискове и ограничения

- **Поддръжка на XSLT:** различните браузъри имат дребни разлики; при нужда може да се генерира статичен HTML с `xsltproc`.

- **Ръчна актуализация:** данните се обновяват ръчно; липсва автоматично синхронизиране с външни източници.
- **Размер на изображенията:** PNG файловете са 1.5–3.2 MB; при бавна мрежа зареждането може да се усеща. Опция: компресия/thumbnail.
- **Валидиране:** пропуск в DOCTYPE/XSD enum води до невалиден @image; checklist в апендикса намалява риска.
- **Производителност при голям обем:** текущият JS филтър/сортиране е напълно клиентски и при стотици записи може да се усеща забавяне; за учебния обем (8–20 реда) е достатъчно.

3.11 Обобщение

Дизайнът демонстрира цялостна XML-първа архитектура: дефиниране и валидиране на структурата, използване на непарснати ентитети за графика, трансформация с XSLT и клиентска интерактивност с JavaScript, оформена с CSS. Връзките между жанрове, компании и игри са строго типизирани чрез key/keyref и ID/IDREF, а изображенията са управлявани през ENTITY декларации.

```

40 </action>
41 </div>
42
43 <table id="gameTable">
44   <thead>
45     <tr>
46       <th onclick="sortable(0, 'string')"/>Title</th>
47       <th onclick="sortable(1, 'date')"/>Release Date</th>
48       <th onclick="sortable(2, 'string')"/>Genre</th>
49       <th onclick="sortable(3, 'string')"/>Platforms</th>
50       <th onclick="sortable(4, 'string')"/>Developer</th>
51       <th onclick="sortable(5, 'string')"/>Publisher</th>
52       <th>Description</th>
53     </tr>
54   </thead>
55
56   <tbody>
57     <xsl-for-each select="games/game">
58       <tr data-game="{@gameid}">
59
60         <td class="title"><xsl-value-of select="title"/></td>
61
62         <!-- Release Date -->
63         <td class="date"><xsl-value-of select="releaseDate"/></td>
64
65         <!-- Genre -->
66         <td class="genre">
67           <xsl-variable name="gid" select="@gameid"/>
68           <xsl-value-of select="..//genres/genre[@id=$gid]/name"/>
69         </td>
70
71         <!-- Platforms -->
72         <td class="platform">
73           <xsl-value-of select="platform"/>
74         </td>
75
76         <!-- Developers -->
77         <td class="developers">
78           <xsl-for-each select="companies/companyref[@role='Developer']">
79             <xsl-if test="position() <= 1"> </xsl-if>
80             <xsl-value-of select="name"/>
81           </xsl-for-each>
82         </td>
83
84         <!-- Publishers -->
85         <td class="publishers">
86           <xsl-for-each select="companies/companyref[@role='Publisher']">
87             <xsl-if test="position() <= 1"> </xsl-if>
88             <xsl-value-of select="name"/>
89           </xsl-for-each>
90         </td>
91
92         <td class="description">
93           <xsl-value-of select="description"/>
94         </td>
95       </tr>
96     </xsl-for-each>
97   </tbody>
98 </table>

```

Фигура 6: XSLT: заглавна част на таблицата и начало на итерацията върху games/game.

```

95         <td class="summary">
96         <cx:value-of select="summary"/>
97         </td>
98
99         <!-- Image -->
100        <td class="image">
101            <img alt="Image" width="200">
102            <cx:attribute name="src">
103                <cx:value-of select="unparsed-entity-uri@image"/>
104            </cx:attribute>
105            </img>
106        </td>
107    </tr>
108    </cx:for-each>
109    </tbody>
110    </table>
111
112    <!-- JavaScript logic -->
113    <script type="text/javascript"><![CDATA[
114
115        function filterTable() {
116            var genre = document.getElementById('genrefilter').value;
117            var search = document.getElementById('search').value.toLowerCase();
118            var table = document.getElementById('gameTable');
119            var rows = table.getElementsByTagName('tbody')[0].getElementsByTagName('tr');
120
121            for (var i = 0; i < rows.length; i++) {
122                var row = rows[i];
123                var title = row.getElementsByTagName('title')[0].textContent.toLowerCase();
124                var genreMatch = (genre === 'all') || (row.getAttribute('data-genre') === genre);
125                var searchMatch = (title.indexOf(search) > -1);
126
127                row.style.display = (genreMatch && searchMatch) ? '' : 'none';
128            }
129        }
130
131        var sortTable = [];
132
133        function sortTable(col, type) {
134            var table = document.getElementById('gameTable');
135            var tbody = table.getElementsByTagName('tbody');
136            var rows = Array.prototype.slice.call(tbody.rows, 0);
137
138            var dir = sortTable[col] === 'asc' ? 'desc' : 'asc';
139            sortTable[col] = dir;
140
141            rows.sort(function(a, b) {
142                var A = a.cells[col].textContent.trim();
143                var B = b.cells[col].textContent.trim();
144
145                if (type === 'date') {
146                    A = new Date(A);
147                    B = new Date(B);
148                } else {
149                    A = A.toLowerCase();
150                    B = B.toLowerCase();
151                }
152            }

```

Фигура 7: XSLT: рендериране на компании, жанрове и описание (продължение).

```

152            }
153
154            if (A < B) return dir === 'asc' ? -1 : 1;
155            if (A > B) return dir === 'asc' ? 1 : -1;
156            return 0;
157        });
158
159        for (var i = 0; i < rows.length; i++) {
160            tbody.appendChild(rows[i]);
161        }
162    }
163    </script>
164    </tbody>
165    </table>
166    </xsl:template>
167    </xsl:stylesheet>
168
169
170

```

Фигура 8: XSLT: долна част на тялото на таблицата и завършване на редовете.

```

110        function filterTable() {
111            var genre = document.getElementById('genrefilter').value;
112            var search = document.getElementById('search').value.toLowerCase();
113            var table = document.getElementById('gameTable');
114            var rows = table.getElementsByTagName('tbody')[0].getElementsByTagName('tr');
115
116            for (var i = 0; i < rows.length; i++) {
117                var row = rows[i];
118                var title = row.getElementsByTagName('title')[0].textContent.toLowerCase();
119                var genreMatch = (genre === 'all') || (row.getAttribute('data-genre') === genre);
120                var searchMatch = (title.indexOf(search) > -1);
121
122                row.style.display = (genreMatch && searchMatch) ? '' : 'none';
123            }
124        }
125
126        var sortTable = [];
127
128        function sortTable(col, type) {
129            var table = document.getElementById('gameTable');
130            var tbody = table.getElementsByTagName('tbody');
131            var rows = Array.prototype.slice.call(tbody.rows, 0);
132
133            var dir = sortTable[col] === 'asc' ? 'desc' : 'asc';
134            sortTable[col] = dir;
135
136            rows.sort(function(a, b) {
137                var A = a.cells[col].textContent.trim();
138                var B = b.cells[col].textContent.trim();
139
140                if (type === 'date') {
141                    A = new Date(A);
142                    B = new Date(B);
143                } else {
144                    A = A.toLowerCase();
145                    B = B.toLowerCase();
146                }
147            }
148
149            if (A < B) return dir === 'asc' ? -1 : 1;
150            if (A > B) return dir === 'asc' ? 1 : -1;
151            return 0;
152        });
153
154        for (var i = 0; i < rows.length; i++) {
155            tbody.appendChild(rows[i]);
156        }
157    }
158
159
160

```

Фигура 9: JavaScript: функции filterTable() и sortTable().

```
1 |
2 | basic styles for the video game catalog.
3 | Defines fonts, colors, and table layout, as well as
4 | styling of filter and search controls.
5 |
6 |
7 | body {
8 |   font-family: Arial, sans-serif;
9 |   margin: 20px;
10 |   background-color: #e0e0e0;
11 |   color: #333;
12 | }
13 |
14 | h1 {
15 |   text-align: center;
16 |   color: #333;
17 | }
18 |
19 | .controls {
20 |   margin-bottom: 10px;
21 |   text-align: center;
22 | }
23 |
24 | .controls label {
25 |   margin-right: 5px;
26 |   font-weight: bold;
27 | }
28 |
29 | .controls select,
30 | .controls input[type="text"],
31 | .controls button {
32 |   padding: 5px 10px;
33 |   margin-right: 10px;
34 |   font-size: 14px;
35 |   height: 30px;
36 |   box-sizing: border-box;
37 | }
38 |
39 | .controls select,
40 | .controls input[type="text"] {
41 |   border: 1px solid #ccc;
42 |   border-radius: 5px;
43 | }
44 |
45 | .controls button {
46 |   cursor: pointer;
47 |   background-color: #fff;
48 |   color: #333;
49 |   border: 1px solid #ccc;
50 |   border-radius: 5px;
51 |   transition: background-color 0.2s ease;
52 | }
53 |
54 | .controls button:hover {
55 |   background-color: #eee;
56 | }
57 |
58 | table {
59 |   width: 100%;
60 |   border-collapse: collapse;
61 |   background-color: #fff;
62 |   box-shadow: 0 0 10px #ccc;
63 | }
```

Фигура 10: CSS: стилове за контролите и табличното оформление.

4. Използване на изкуствен интелект

Изкуственият интелект беше използван целенасочено и ограничено: (1) помощ при формулиране/редакция на текстови части и идеи за оформление; (2) генериране на неутрални изображения за игрите чрез text-to-image промптове (без фактическо съдържание и търговски марки). Архитектурата, XML моделирането, XSD, XSLT, JS и данните са разработени ръчно; ИИ не е използван за логика или структура.

Генериране на изображения: използвани са кратки промптове с контрол върху стил и цветове, целящи универсални икони/визуализации. Оценка: яснота, липса на подвеждащи детайли, съвместимост с дизайна. Всички файлове минават през ръчна проверка и XSD валидация; ИИ е помощен инструмент, а коректността остава отговорност на авторите.

5. Тестване

Среда: Chrome/Edge/Firefox (актуални версии), локален прост HTTP сървър при нужда. Сценарии: сортиране по дата (ISO), търсене с главни/малки букви, филтър по жанр, проверка на счупен ентитет (очаквана грешка при валидация), липсващ жанр (очакван отказ от XSD), коректно парсиране на дати, липса на счупени `src` за изображения, отсъствие на грешки в конзолата.

5.1 Валидация на XML съдържанието

Команда: `xmllint -schema catalog.xsd catalog.xml`. Проверяват се задължителни елементи/атрибути, ID/IDREF, key/keyref и валидни enum стойности за `image`. Основната проверка е локална; при нужда може и онлайн валидатор.

5.2 Тестване на XSLT трансформацията

Потвърдено е, че `catalog.xsl` обхожда всички игри, резолвира жанрове и компании, визуализира атрибутите (`releaseDate`, `genreId`, `image`) и генерира нужните HTML атрибути (напр. `data-genre`). Резултатният HTML отразява данните точно.

5.3 Тестване на визуализацията и стилизирането

В Chrome/Firefox/Edge е проверено подравняването на таблицата, четимостта на текстовете, зареждането на изображенията и консистентността на интерфейса при различни размери на прозореца.

5.4 Тестване на интерактивната функционалност

Ръчно е проверено: филтър по жанр, търсене по заглавие (case-insensitive), сортиране по заглавие/дата, обновяване на редовете без презареждане и липса на JS грешки в конзолата.

5.5 Обобщение на резултатите

Компонентите работят коректно самостоятелно и заедно: XML е валиден, XSLT трансформацията е точна, визуализацията е стабилна, интерактивността функционира в тестовите браузъри.

Автоматизация (примерни команди):

```
make validate      # xmllint --noout --schema catalog.xsd catalog.xml
make html          # xsltproc catalog.xsl catalog.xml > build/output.html
make serve         # python3 -m http.server 8000
```

Обобщение на изпълнените тестове:

Категория	Сценарий	Резултат
Валидация	<code>xmllint -schema catalog.xsd catalog.xml</code>	Успешно
Функционалност	Търсене по подниз, филтър по жанр, сортиране по дата/заглавие	ОК (Chrome/Edge/Firefox)
Медии	Зареждане на 8 PNG през <code>ENTITY + unparsed-entity-uri()</code>	Всички визуализирани
Негативни	Грешен <code>genreId</code> и липсващ <code>companyRef</code> (очаквана невалидност)	Успешно засечени

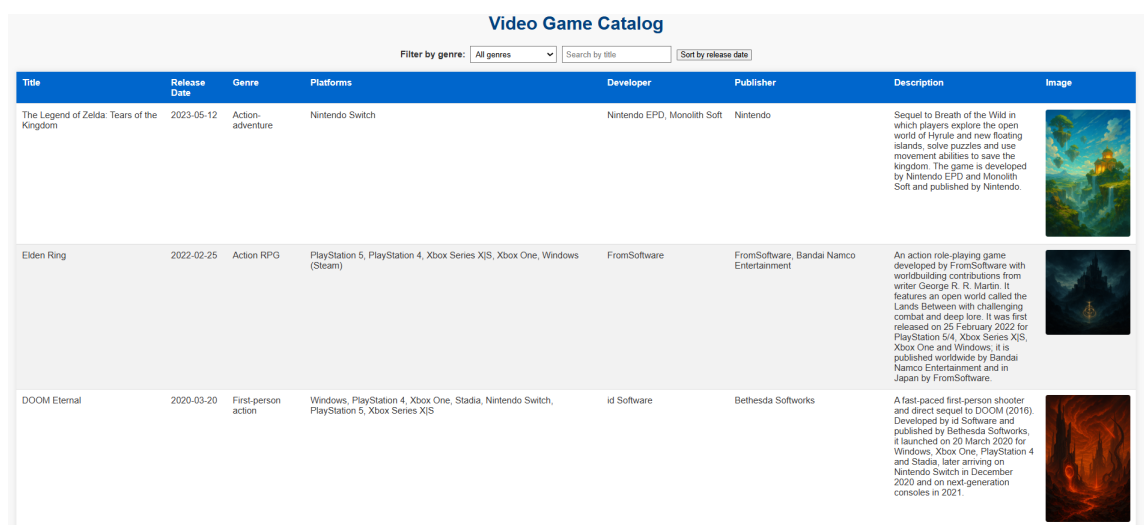
Негативните тестове се изпълняват ръчно чрез временно въвеждане на грешни стойности (несъществуващ жанр/компания, празно `companyRef`, невалидна дата), след

което `xmllint` трябва да прекрати с грешка. Във визуалния слой се проверява, че при липсващо изображение XSD блокира стойността на атрибут `image`.




Негативни тестове (примерни) и очакван резултат:

Сценарий	Стъпка	Очакван резултат
Липсващ жанр	<code>genreId="g999"</code>	<code>keyref gameGenreRef</code> грешка при <code>xmllint</code>
Несъществуваща компания	<code>companyRef name="Foo" role="developer"</code>	<code>keyref companyRefKey</code> грешка
Невалидно изображение	<code>image="halo"</code>	<code>cvc-enumeration-valid</code> грешка
Невалидна дата	<code>releaseDate="2023/12/01"</code>	<code>cvc-datatype-valid.1.2.1</code> за <code>xs:date</code>

Бъдещи тестове (ако се разширява): добавяне на mock данни с >50 записа за оценка на производителността на клиентските филтри; проверка на адаптивността на CSS на потесни екрани; тест с компресирани изображения за сравнение на време за зареждане; unit тест за XSD с умишлено счупени стойности (грешен `genreId`, липсващ `company`, невалидна дата).



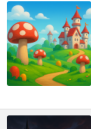
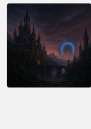
The screenshot shows a web application titled "Video Game Catalog". At the top, there are filters: "Filter by genre: All genres" (a dropdown menu), a "Search by title" input field, and a "(Sort by release date)" link. Below these is a table with the following columns: Title, Release Date, Genre, Platforms, Developer, Publisher, Description, and Image. The table lists three games: "The Legend of Zelda: Tears of the Kingdom", "Elden Ring", and "DOOM Eternal". Each row includes a small image of the game cover.

Title	Release Date	Genre	Platforms	Developer	Publisher	Description	Image
The Legend of Zelda: Tears of the Kingdom	2023-05-12	Action-adventure	Nintendo Switch	Nintendo EPD, Monolith Soft	Nintendo	Sequel to Breath of the Wild in which players explore the open world of Hyrule and new floating islands, solve puzzles and use movement abilities to save the kingdom. The game is developed by Nintendo EPD and Monolith Soft and published by Nintendo.	
Elden Ring	2022-02-25	Action RPG	PlayStation 5, PlayStation 4, Xbox Series X/S, Xbox One, Windows (Steam)	FromSoftware	FromSoftware, Bandai Namco Entertainment	An action role-playing game developed by FromSoftware with worldbuilding contributions from writer George R. R. Martin. It features an open world called the Lands Between with challenging combat and deep lore. It was first released on 25 February 2022 for PlayStation 5/4, Xbox Series X/S, Xbox One and Windows. It is published worldwide by Bandai Namco Entertainment and in Japan by FromSoftware.	
DOOM Eternal	2020-03-20	First-person action	Windows, PlayStation 4, Xbox One, Stadia, Nintendo Switch, PlayStation 5, Xbox Series X/S	id Software	Bethesda Softworks	A fast-paced first-person shooter and direct sequel to DOOM (2016). Developed by id Software and published by Bethesda Softworks, it launched on 20 March 2020 for Windows, Xbox One, PlayStation 4 and Stadia, later arriving on Nintendo Switch in December 2020 and on next-generation consoles in 2021.	

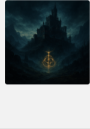

Фигура 11: Преглед на каталога с активни контроли (начален изглед).

Minecraft	2011-11-18	Sandbox / Survival	Windows, macOS, Linux, Xbox, PlayStation, Nintendo Switch, mobile devices	Mojang Studios	Mojang Studios	A sandbox game with survival elements, originally created by Markus Persson and later developed by Mojang Studios. The official version was released on 18 November 2011 after a public alpha starting in 2009. Players can freely explore and build in a procedurally generated blocky world, gather resources, craft items and try to survive. It is both developed and published by Mojang Studios and is the best-selling video game in history.	
Tetris	1984-06-06	Puzzle	Elektronika 60, IBM PC, Amiga, Atari ST, Apple II, NES, Game Boy and many others	Alexey Pajitnov, Vadim Gerasimov	The Tetris Company	A classic puzzle game created by Alexey Pajitnov in the Soviet Union, with the MS-DOS version programmed by Vadim Gerasimov. Colored tetromino blocks fall from the top and players rotate and position them to form complete lines. First released on 6 June 1984, it has appeared on numerous platforms and is considered one of the most influential puzzle games of all time.	
The Witcher 3: Wild Hunt	2015-05-19	Action RPG	Windows, PlayStation 4, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S	CD Projekt RED	CD Projekt	An open-world action role-playing game developed by CD Projekt RED. Players control Geralt of Rivia, a monster hunter searching for his adopted daughter who is pursued by the spectral Wild Hunt. The game launched on 19 May 2015, came to Nintendo Switch in October 2019 and received next-generation versions in December 2022. It is published by CD Projekt.	

Фигура 12: Средна част на таблицата (редове 4–7).

Super Mario Odyssey	2017-10-27	Platformer	Nintendo Switch	Nintendo EPD, 1-Up Studio	Nintendo	A 3D platformer developed by Nintendo EPD Tokyo and 1-Up Studio and published by Nintendo. Mario travels across many kingdoms with his new friend, the sentient hat Cappy, collecting Power Moons to fuel the airship Odyssey. The game was released on 27 October 2017 and later in additional Asian regions.	
Baldur's Gate 3	2023-08-03	Role-playing	Windows, PlayStation 5, macOS, Xbox Series X/S, Linux	Larian Studios	Larian Studios	A role-playing game based on the Dungeons & Dragons ruleset, developed and published by Larian Studios. Players explore the world of the Forgotten Realms, create characters and make story-defining choices. The full Windows version released on 3 August 2023, followed by PlayStation 5 in September 2023, macOS in September 2023, Xbox Series X/S in December 2023 and a planned Linux version in 2025.	


Фигура 13: Долен край на таблицата (последен ред).

Video Game Catalog							
Filter by genre: Action RPG <input type="text" value="Search by title"/> Sort by release date							
Title	Release Date	Genre	Platforms	Developer	Publisher	Description	Image
Elden Ring	2022-02-25	Action RPG	PlayStation 5, PlayStation 4, Xbox Series X/S, Xbox One, Windows (Steam)	FromSoftware	FromSoftware, Bandai Namco Entertainment	An action role-playing game developed by FromSoftware with worldbuilding contributions from writer George R. R. Martin. It features an open world called the Lands Between with challenging combat and deep lore. It was first released on 25 February 2022 for PlayStation 5/4, Xbox Series X/S, Xbox One and Windows. It is published worldwide by Bandai Namco Entertainment and in Japan by FromSoftware.	
The Witcher 3: Wild Hunt	2015-05-19	Action RPG	Windows, PlayStation 4, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S	CD Projekt RED	CD Projekt	An open-world action role-playing game developed by CD Projekt RED. Players control Geralt of Rivia, a monster hunter searching for his adopted daughter who is pursued by the spectral Wild Hunt. The game launched on 19 May 2015, came to Nintendo Switch in October 2019 and received next-generation versions in December 2022. It is published by CD Projekt.	

Фигура 14: Филтър по жанр “Action RPG”.

Video Game Catalog





Filter by genre:

Title	Release Date	Genre	Platforms	Developer	Publisher	Description	Image
Super Mario Odyssey	2017-10-27	Platformer	Nintendo Switch	Nintendo EPD, 1-Up Studio	Nintendo	A 3D platformer developed by Nintendo EPD Tokyo and 1-Up Studio and published by Nintendo. Mario travels across many kingdoms with his new friend, the sentient hat Cappy, collecting Power Moons to fuel the airship Odyssey. The game was released on 27 October 2017 and later in additional Asian regions.	

Фигура 15: Търсене по заглавие “mario”.

Video Game Catalog

Filter by genre:

Title	Release Date	Genre	Platforms	Developer	Publisher	Description	Image
Tetris	1984-06-06	Puzzle	Elektronika 60, IBM PC, Amiga, Atari ST, Apple II, NES, Game Boy and many others	Alexey Pajitnov, Vadim Gerasimov	The Tetris Company	A classic puzzle game created by Alexey Pajitnov in the Soviet Union, with the MS-DOS version programmed by Vadim Gerasimov. Colored tetrimino blocks fall from the top and players rotate and position them to form complete lines. First released on 6 June 1984, it has appeared on numerous platforms and is considered one of the most influential puzzle games of all time.	
Minecraft	2011-11-18	Sandbox / Survival	Windows, macOS, Linux, Xbox, PlayStation, Nintendo Switch, mobile devices	Mojang Studios	Mojang Studios	A sandbox game with survival elements, originally created by Markus Persson and later developed by Mojang Studios. The official version was released on 18 November 2011 after a public alpha starting in 2009. Players can freely explore and build in a procedurally generated blocky world, gather resources, craft items and try to survive. It is both developed and published by Mojang Studios and is the best-selling video game in history.	
The Witcher 3: Wild Hunt	2015-05-19	Action RPG	Windows, PlayStation 4, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S	CD Projekt RED	CD Projekt	An open-world action role-playing game developed by CD Projekt RED. Players control Geralt of Rivia, a monster hunter searching for his adopted daughter who is pursued by the spectral Wild Hunt. The game launched on 19 May 2015, came to Nintendo Switch in October 2019 and received next-generation versions in December 2022. It is published by CD Projekt.	
Super Mario Odyssey	2017-10-27	Platformer	Nintendo Switch	Nintendo EPD, 1-Up Studio	Nintendo	A 3D platformer developed by Nintendo EPD Tokyo and 1-Up Studio and published by Nintendo. Mario travels across many kingdoms with his new friend, the sentient hat Cappy.	

Фигура 16: Сортиране по дата (низходящо).

6. Заключение и възможно бъдещо развитие

Проектът демонстрира как чрез XML и свързани технологии може да се реализира богат, интерактивен каталог без нужда от бекенд. Предимствата са ясната структура, автоматичната валидация, лесната поддръжка и преносимост. Основните ограничения са ръчната актуализация и зависимостта от XSLT поддръжка в браузърите.

В бъдеще каталогът може да се разшири с оценки и ревюта, филтри по платформи и времеви интервали, локализация на интерфейса, генериране на PDF чрез XSL-FO и по-строга типизация (например отделна колекция за платформи). Други идеи включват йерархия на жанровете, интеграция с външни API за автоматично обновяване на данните и версия на мобилно приложение. Планирани разширения: поддръжка на поджанрове и PEGI/ESRB рейтинги; филтри по платформа и диапазон дати; PDF изход чрез XSL-FO; локализация на интерфейса; по-строга типизация на платформи

чрез отделен `enum`/колекция; интеграция с външни API за автоматично обновяване на каталога. Практическа приложимост: подходът е подходящ за малки/средни каталози, офлайн справочници и учебни демонстрации. Поддръжка: при обновяване се следва `checklist` от апендикса (данни \rightarrow `DOCTYPE/XSD` \rightarrow `xmllint` \rightarrow визуален преглед \rightarrow по желание `xsltproc`). При нужда от по-мощно решение може да се въведе бекенд или API синхронизация, но текущият файл-базиран модел остава лек и преносим.

Критерии за приемане:

- `catalog.xml` валиден срещу `catalog.xsd` (`xmllint` без грешки).
- Всички изображения зареждат коректно чрез `ENTITY` и `unparsed-entity-uri()`.
- Търсене, филтър по жанр и сортиране работят в поне два браузъра.

7. Разпределение на работата

Работата по проекта беше разделена така:

#	Дейност	Отговорник
1	Събиране и проверка на данни (игри, жанрове, платформи, компании)	Ивайло Кънчев
2	Проектиране на XML структура и XSD (ключове, <code>keyref</code> , <code>enum</code> за изображения)	Ивайло Кънчев
3	Разработка на XSLT и клиентски JS (сортиране, филтър, търсене)	Йоан Байчев
4	CSS стилове и визуална структура	Йоан Байчев
5	Подготовка на изображения, <code>ENTITY</code> в <code>DOCTYPE</code> , синхрон с XSD <code>enum</code>	Ивайло Кънчев, Йоан Байчев
6	Тестове (валидация, функционални, визуални)	Ивайло Кънчев, Йоан Байчев
7	Документация и финален преглед	Ивайло Кънчев, Йоан Байчев

8. Източници и литература

Използвани бяха официални сайтове на компаниите (Nintendo, FromSoftware, Bethesda, Mojang, Larian и др.), енциклопедични източници за проверка на дати и платформи (например Wikipedia), спецификациите на W3C за XML, XML Schema и XSLT, документацията на `libxml2/xmllint`, както и курсови материали по „XML технологии за семантичен Уеб“. Използваните източници:

- [1] W3C XML 1.0 Recommendation. Достъпно на: <https://www.w3.org/TR/xml/>.
- [2] W3C XML Schema Part 1: Structures. Достъпно на: <https://www.w3.org/TR/xmlschema-1/>.
- [3] W3C XSL Transformations (XSLT) 1.0. Достъпно на: <https://www.w3.org/TR/xslt>.

- [4] libxml2/xmllint Documentation. Достъпно на: <http://xmlsoft.org/>.
- [5] Материали по „XML технологии за семантичен Уеб“ (Moodle на курса).
- [6] Wikipedia статии за датите и платформите на игрите: Elden Ring, DOOM Eternal, Minecraft, Tetris, The Witcher 3, Super Mario Odyssey, Baldur's Gate 3.
- [7] libxslt/xsltproc Documentation. Достъпно на: <http://xmlsoft.org/xslt.html>.
- [8] MDN Web Docs: XSLT in the Browser (XSLTProcessor, transformToDocument). Достъпно на: <https://developer.mozilla.org/en-US/docs/Web/API/XSLTProcessor>.
- [9] Liquid Technologies Online XSD Validator (използван за допълнителна проверка на catalog.xml и catalog.xsd). Достъпно на: <https://www.liquid-technologies.com/online-xsd-validator>.

9. Апендикс

- **Файлова структура** – catalog.xml, catalog.xsd, catalog.xsl, style.css, директории images/ (осем PNG) и screenshots/, documentation/documentation.tex.
- **Валидация:** `xmllint -noout -schema catalog.xsd catalog.xml`.
- **Трансформация извън браузър:** `xsltproc catalog.xsl catalog.xml > output.html`.
- **Роля на файловете:** catalog.xml (данни), catalog.xsd (схема и ограничения), catalog.xsl (трансформация + JS), style.css (визуален стил), images/ (медии), documentation/documentation.tex (документация).
- **Workflow checklist:** добави/актуализирай данни в XML; обнови DOCTYPE/XSD enum при ново изображение; пусни xmllint; провери визуално в браузър; при нужда генерирай статичен HTML с xsltproc; при много нови изображения помисли за компресия/thumbnail.
- **Негативни проверки:** временно въведи грешен genreId или image и очаквай отказ от xmllint; върни коректните стойности след теста.
- **Полезни команди:**
 - `xmllint -schema catalog.xsd catalog.xml` – валидация.
 - `xsltproc catalog.xsl catalog.xml > build/output.html` – статичен HTML.
 - `python3 -m http.server 8000` – локален сървър за преглед.
- **Допълнителни материали:** екранни снимки, инструкции за локално отваряне (`python -m http.server 8000` и `http://localhost:8000/catalog.xml`) и др.
- **Локално стартиране без риск за файловата система:**
 - VS Code Live Server: инсталирай разширението, отвори папката, натисни Go Live, зареди `http://localhost:5500/catalog.xml`.

- Python HTTP server: `python3 -m http.server 8000`, после `http://localhost:8000/catalog.xml`.

Това елиминира ограниченията за достъп до локални ресурси в съвременните браузъри.

- **Бележка за съвместимост:** Ако се ползва остарял IE/IE mode, може да се отвори трансформацията директно от файловата система, но се препоръчва модерен браузър + локален сървър за пълна функционалност на JS/сортиране.