



**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 № 22:**

**„Каталог за Human Resource Manager - 2“**

**Автори:**

Борил Игнатов, Ф№ 0MI0600124

Гена Генова, Ф№ 9MI0600073

**Ръководители:**

доц. д-р Александър Димов

гл. ас. д-р Явор Данков

януари, 2024

София

## Съдържание

1	Въведение .....	3
2	Анализ на решението .....	4
2.1	Работен процес .....	4
2.2	Структура на съдържанието .....	5
2.3	Тип и представяне на съдържанието .....	7
3	Дизайн .....	9
4	Тестване .....	15
5	Заклучение и възможно бъдещо развитие .....	17
6	Разпределение на работата .....	18
7	Използвани литературни източници и Уеб сайтове .....	19
8	Апендикс .....	19

# 1 Въведение

Целта на проекта е да се представи каталог на Human Resource Manager (HRM) система, поддържащ информация за служителите в дадена компания по групи в зависимост от техните рангове, типове (пълен/непълен трудов договор, стажанти, на хонорар-сметка, и др.) и други характеристики (лични данни, заплата, стаж, компетенции, атестирания, ....). Каталогът представя графичното съдържание чрез XML entities. Връзките в каталога – напр. между групи и подгрупи, между подгрупи и продукти, и др. – се описват чрез (съставни) ключове и референции към тях. Проектът представя информация за описанието на 7-8 служителя и се валидира чрез подходящо изграден за целта XML Schema документ. Създаденото XML съдържание може да се представи в HTML браузер чрез CSS и XSLT, с възможност за сортиране на представените обекти по няколко от техните признаци.

- Актуалността на този проблем е от огромно значение, защото целта да се създаде лесно достъпна и използвана система, изобразяваща каталог от служителите от някаква фирма се търси от всички. Начинът на презентирание на служителите и техните началници би направило огромно значение в презентирането на самата фирма, защото външният вид в корпоративния свят е печеливш в много страни, надпреварващи се с конкуренцията си.
- Проблемът, който разрешава проектът, е фокусиран върху правилното структуриране на данните на служителите и проблемите с разпределението, съхранението и визуализацията на информацията.
- Проектът използва XML технологии за създаване на HRM каталог, където данните за служителите са групирани и организирани. Информацията се представя графично чрез XML entities, а връзките между групите и подгрупите се установяват с ключове и референции. XML Schema документ се използва за валидация на данните, осигурявайки тяхната консистентност и точност.
- Документацията разполага със следната структура:
  - Въведение
    - Обобщение на целите и обхвата на документа.
  - Анализ на решението
    - В този раздел се извършва анализ на решението, фокусирайки се върху различни аспекти. Описва се работният процес, който е част от решението. Този раздел обяснява как се осъществява създаването и управлението на HRM каталога с използване на XML технологии. Подчертава се структурата на съдържанието в HRM каталога. Обяснява се как информацията за служителите е организирана, като се дава представа за групирането по рангове, типове трудови договори и други характеристики.

- Дизайн
  - В раздела за дизайн се обсъжда архитектурата и външният вид на разработения HRM каталог. Включва се информация за използваните CSS и XSLT технологии за представяне на XML съдържание в HTML браузер.
- Тестване
  - Разделът за тестване описва процеса на проверка на функционалността и коректността на разработения HRM каталог. Подчертава се валидацията на данните чрез XML Schema документ и възможността за сортиране на обектите в браузера.
- Заключение и възможно бъдещо развитие
  - В заключението се обобщават постигнатите резултати и се предоставят възможни насоки за бъдещо развитие на проекта. Анализира се успешната реализация и идентифицират се възможности за подобрения.
- Разпределение на работата
  - Този раздел описва как работата е разпределена между членовете на екипа. Подчертават се индивидуалните отговорности и приносът на всеки участник към реализацията на проекта.
- Използвани литературни източници и Уеб сайтове
  - Предоставят се използваните литературни източници и уеб сайтове, които са консултирани при изготвянето на проекта. Обозначават се източниците, които са допринесли за успешната реализация.
- Апендикс
  - В апендикса се прикачват допълнителни материали, като кодове, схеми, изображения или други ресурси, които подпомагат разбирането и допълват информацията в документа.

## 2 Анализ на решението

### 2.1 Работен процес

Проектът представлява съвременна система за управление на данните за служителите и отделите в организацията. За да осигурим оптимална структура и лесен достъп до информацията, използваме XML формат за представяне на данните за всеки служител. Всяка единица от този формат включва информация за лични данни, заетост, местоживеене и отдел.

Един от ключовите елементи в проекта е използването на XSLT стилове за трансформация на XML данните в HTML формат. Тези стилове определят визуалния вид на уеб интерфейса, който позволява лесно визуализиране в уеб браузъри. Системата ни гарантира динамичен и структуриран интерфейс, предоставящ информацията на потребителите по оптимален начин. Спомагателният файл към XSL файла е CSS файл, който задава лика на уебсайта, който се определя от цветовете, размерността на представените данни от XML документа, удобният потребителски интерфейс.

Резултатът от този проект е HTML документ, който представлява уеб интерфейс за бърз и удобен достъп до информацията. Този интерфейс предоставя сортирани и структурирани данни за служителите и отделите, като запазва визуалната цялост и предоставя проста навигация. Потребителите могат лесно да преглеждат, филтрират и сортират информацията, което ги прави управляеми и подготвени за вземане на решения.

Създаването на този проект не само оптимизира процеса на управление на данните, но и предоставя инструмент за по-ефективно вземане на решения в организацията. Нашата система е създадена с фокус върху потребителското удобство и възможностите за персонализация, като това го прави необходима част от работните процеси във всяка организация.

## 2.2 Структура на съдържанието

В структурата на XML съдържанието, данните за служителите са организирани във възможно най-ясна и логична структура. Използвана е таксономия, която обхваща основните категории от лични данни, данни за заетостта и адресна информация. Основният коренов елемент в XML документа е catalogue, от който се наследяват елементите employeesList и departmentsList. Наследникът на employeesList е employee, който очертава точно един служител в XML файлът, а departmentsList има наследник department, който очертава точно едно отделение във фирмата. Елементите employee и department служат, за да зададат границите и дефинициите между всеки един потребител и всяко едно отделение в каталога. По-долу са отбелязани в скоби и подчертани типовете данни за всеки един XML елемент според schema.xsd:

Employee се наследява от следните няколко елемента :

1. personData (личните данни на служителя). Наследява се от:

- firstName: Име (xs:string)
- middleName: Презиме (xs:string)
- lastName: Фамилия (xs:string)
- id: Идентификационен номер (уникален в самата фирма) (xs:string)
- birthday: Дата на раждане (xs:date, зададен като dateType)
- telephoneNumber: Телефонен номер (xs:string, зададен по регулярен израз \+\d{1,})
- residenceData: Информация за местоживееене

- country: Държава (xs:string)
- city: Град (xs:string)
- zipCode: Пощенски код (xs:string)
- address: Адрес (xs:string)

2. employeeData (данни за работната заетост на служителя). Наследява се от:

- departmentName: Име на отдел (xs:string)
- workweek: Работна седмица (пълен/непълен работен ден) (xs:string)
- salary: Заплата (xs:decimal)
- employmentType: Вид на заетостта (постоянна, стажант, и т.н.)
- positionInCompany: Позиция в компанията
  - positionName: Име на позицията в компанията (xs:string)
  - startDate: Начална дата (xs:date, зададен като dateType)

Взаимовръзки и Характеристики:

- Лични Данни (personData):
  - *Честота на срещане:* Всеки служител във файла представлява единичен екземпляр от тази категория.
  - *Взаимовръзки:* Връзката с residenceData уточнява информацията за адреса на служителя.
- Данни за Заетостта (employeeData):
  - *Честота на срещане:* Всеки служител има само една категория от този тип.
  - *Взаимовръзки:* Връзката с positionInCompany конкретизира текущата длъжност и началната дата на заетост.

Кореновият елемент catalogue съдържа ключ (managerKey) и референция към него (managerRef), който се използва да селектира личните данни на потребителя, като по този начин ще се изобразят на уеб браузера firstName, lastName и id. Същата аналогия изпълнява managerRef само че се прилага за firstNameRef, lastNameRef и idRef.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="stylesheet.xsl"?>
<catalogue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="schema.xsd">
  <employeesList>
    <employee>
      <personData>
        <firstName>Boril</firstName>
        <middleName>Georgiev</middleName>
        <lastName>Stoyanov</lastName>

        <id>14214214</id>
        <birthday>1999-10-05</birthday>
        <telephoneNumber>+359473829109</telephoneNumber>

        <residenceData>
          <country>Bulgaria</country>
          <city>Sofia</city>
          <zipCode>1001</zipCode>
          <address>3 Yoan Ekzarh Street</address>
        </residenceData>
      </personData>

      <employeeData>
        <departmentName>Backend</departmentName>
        <workweek>fullTime</workweek>
        <salary>2500.00</salary>
        <employmentType>permanent</employmentType>

        <positionInCompany>
          <positionName>Junior Developer</positionName>
          <startDate>2023-05-24</startDate>
        </positionInCompany>
      </employeeData>
    </employee>
  </employeesList>
  <departmentsList>
    <department>
      <departmentName>Frontend</departmentName>
      <manager firstNameRef="Kostadin" lastNameRef="Dimitrov" idRef="14214220"/>
    </department>
    <department>
      <departmentName>Backend</departmentName>
      <manager firstNameRef="Dimitar" lastNameRef="Yavorov" idRef="14214221"/>
    </department>
    <department>
      <departmentName>Admin</departmentName>
      <manager firstNameRef="Plamena" lastNameRef="Aleksandrova" idRef="14214218"/>
    </department>
  </departmentsList>
</catalogue>

```

Фиг. 1. – Основно съдържание на XML документа (игнорирано DTD, понеже е коментирано)

## 2.3 Тип и представяне на съдържанието

Съдържанието на каталога е представено през Internet Explorer браузера. HTML файлът се състои от текст, представляващ подробна информация за служителите в съответната организация, изображения, които са 9 наброй. Всеки служител разполага с една снимка, която се дефинира от неговият уникален идентификационен номер, който се намира в съответната секция в XML документа.

Текстово Съдържание:

### 1. XML документ:

- Брой: 1
- Описание: XML файл, който съдържа данни за служителите и отделите в организацията.
- Източник: Вътрешно създаден.

- Размер: Приблизително 9 KB.
- Кодиране: UTF-8.

Графично Съдържание:

1. Изображения на служители:

- Брой: 9 (по едно за всеки служител)
- Описание: Снимки на служителите, използвани за визуално представяне в стил на картички в интерфейса.
- Източник: Външно (посочено в 6-та точка)
- Размер и Тип на Файла: JPG формат, с размери 80x80 пиксела (автоматично пасване на изображението в HTML документа). Всеки файл заема различно място в хранилището.
- Кодиране: Няма кодиране, тъй като става въпрос за бинарни изображения.

Допълнителни файлове:

1. XSL документ:

- Брой: 1 (stylesheet.xsl)
- Описание: Файл, който дефинира стиловете и скриптовете за преобразуване на XML данните в HTML за представяне в интерфейса.
- Източник: Вътрешно създаден.
- Тип на Файла: Текстов файл с разширение .xsl.
- Размер: 5 KB.
- Кодиране: UTF-8.

2. XSD документ:

- Брой: 1 (schema.xsd)
- Описание: Файл, който дефинира границите, структурата и типовете данни на елементите в XML файла (data.xml).
- Източник: Вътрешно създаден.
- Тип на Файла: Текстов файл с разширение .xsd.
- Размер: Приблизително 4 KB.
- Кодиране: UTF-8.



## 2. CSS файл

- Брой: 1 (style/style.css)
- Описание: Файл, който задава графичният вид на HTML уебсайт при трансформацията от XML, използвайки XSL файла stylesheet.xsl.
- Източник: Вътрешно създаден.
- Тип на Файла: Текстов файл с разширение .css.
- Размер: Приблизително 1.7 KB.
- Кодирание: Не е зададено. Подразбира се UTF-8.

Този подход гарантира ефективна обработка на данните и оптимизирано представяне в потребителския интерфейс през Internet Explorer.

## 3 Дизайн

Решението на проблема е пряко изпълнимо с изискванията, написани в условието, като включва:

- XML 1.0;
- XSD 1.0;
- XSL 1.0;
- CSS документ, който оформя графично съдържанието на XML документа при визуализация от браузера.
- Браузер, на който да се генерира графично съдържанието от XML документа – Internet Explorer.

Проектът презентира каталог на служителите от една IT фирма (не е специфицирано в условието на заданието каква трябва да бъде фирмата). Структурата на данните, които са представени през съответния браузер при изпълнението на XML документа, осъществява съответните изисквания като включва рангове, типове пълен/непълен трудов договор, стаж, на хонорар-сметка, лични данни, заплата, стаж, компетенции, атестирания, които са представени под формата на „кутии“, като броят кутии се определя от броят разделения/подгрупи <employee> в наследената група <employeesList>. Проектът използва XML технологии за структуриране, представяне и обработка на информацията за служителите и отделите в организацията. В следващите подраздели ще бъдат обяснени архитектурните решения, използвани технологии и валидацията на данните.

Архитектура на Решението:

1. Структура на Директорията:

- data.xml: Основен XML документ, съдържащ данни за служителите и отделите.
- stylesheet.xsl: Файл с XSL стилове и скриптове за трансформация на XML в HTML.
- style.css: Файл с CSS стилове за оформление на HTML.
- schema.xsd: Файл с XSD, служещ за валидация на данните в XML документа.

## 2. Схема на Решението:

- Потребителски Интерфейс: HTML интерфейс, генериран от XSLT преобразуване.
- Бизнес Логика: XML документ с данни за служителите и отделите.
- Визуализация: Използване на XSL стилове за представяне на данните в удобен и лесно разбираем вид.

Използвани технологии:

### 1. XML и XML Schema:

- Използване на XML за структуриране на информацията.
- Въведено XML Schema (schema.xsd) за дефиниране на структурата и валидация на данните.

### 2. XSL (stylesheet.xsl):

- Определени са XSL стилове, които представят XML данните в удобен и интуитивен HTML вид.
- Скриптове в XSLT позволяват взаимодействие и превключване между различни изгледи (служители и отдели).

### 3. CSS (style.css):

- Стилите са дефинирани в CSS, което подобрява визуалния аспект на HTML страницата.

Използване на XML Entities и Валидация:

### 1. XML Entities:

- Използвани са връзки чрез атрибути и референции, като например връзката между служител и отдел чрез атрибутите firstNameRef, lastNameRef, и idRef.

### 2. XSD (schema.xsd):

- Дефинирана е XML Schema (XSD) за валидация на структурата на XML документа.
- Схемата съдържа определения за различните типове данни и изискванията за тяхната валидация.

Използвано е за валидиране на XML документа (data.xml) чрез външен XSD документ (schema.xsd) посредством сайта: <https://www.xmlvalidation.com/>.

XSD документът съдържа декларациите на елементите и техните зададени ограничения и типове данни, които ги дефинират. Да разгледаме следното парче код от файла schema.xsd:

```
<!-- Define types -->
<xs:simpleType name="dateType">
  <xs:restriction base="xs:date"/>
</xs:simpleType>

<xs:simpleType name="decimalType">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>

<xs:simpleType name="phoneType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\+\d{1,}"/>
  </xs:restriction>
</xs:simpleType>
```

Кодът дефинира нови типове данните dateType, decimalType и phoneType. Като ще разгледаме всеки един сегмент по-долу.

```
<xs:simpleType name="dateType">
  <xs:restriction base="xs:date"/>
</xs:simpleType>
```

Този код дефинира прост тип данни с име dateType, който е базиран на вградения тип xs:date в XML Schema. Този тип се използва за представяне на дати във формат, съответстващ на стандарта ISO 8601 за дати.

```
<xs:simpleType name="decimalType">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

Този код дефинира прост тип данни с име decimalType, който е базиран на вградения тип xs:decimal в XML Schema. Този тип се използва за представяне на числа с плаваща запетая.

```
<xs:simpleType name="phoneType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\+\d{1,}"/>
  </xs:restriction>
</xs:simpleType>
```

Този код дефинира прост тип данни с име `phoneType`, който е базиран на вградения тип `xs:string` в XML Schema. Този тип има ограничение, което се постига с елемента `<xs:restriction>`. Ограничението в този случай е `xs:pattern`, което уточнява, че стойността на този тип трябва да отговаря на определен регулярен израз. В този конкретен случай, регулярният израз `\+\d{1,}` означава, че стойността трябва да започва със знак плюс (+) и след това да следват поне една или повече цифри. Така се спазва обичайната практика за представяне на международни телефонни номера.

Сега да разгледаме дефинираните елементи в `xsd` файла:

```
<xs:complexType name="personDataType">
  <xs:sequence>
    <xs:element name="firstName" type="xs:string"/>
    <xs:element name="middleName" type="xs:string"/>
    <xs:element name="lastName" type="xs:string"/>
    <xs:element name="id" type="xs:string"/>
    <xs:element name="birthday" type="dateType"/>
    <xs:element name="telephoneNumber" type="phoneType"/>
    <xs:element name="residenceData" type="residenceDataType"/>
  </xs:sequence>
</xs:complexType>
```

Под-елементите `firstName`, `middleName`, `lastName`, `id`, `birthday`, `telephoneNumber` и `residenceData` определят информацията за физическото лице. Всяко физическо лице има име (първо, средно и последно), идентификационен номер (`id`), дата на раждане (`birthday`), телефонен номер (`telephoneNumber`) и данни за пребиваване (`residenceData`). Типовете на елементите, като `xs:string`, `dateType`, и `phoneType`, указват какви видове стойности се очакват за съответните атрибути.

```
<xs:complexType name="residenceDataType">
  <xs:sequence>
    <xs:element name="country" type="xs:string"/>
    <xs:element name="city" type="xs:string"/>
    <xs:element name="zipCode" type="xs:string"/>
    <xs:element name="address" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Под-елементите `country`, `city`, `zipCode` и `address` дефинират информация за пребиваване или адрес. Всяка една от тях е от тип `xs:string`, което означава, че те очакват символни низове (текст). Елементът `<xs:sequence>` указва, че под-елементите (`country`, `city`, `zipCode`, `address`) трябва да се появяват в определен ред, както е посочено в документа. Този ред е важен при валидацията на XML документите срещу този XSD.

```

<xs:complexType name="employeeDataType">
  <xs:sequence>
    <xs:element name="departmentName" type="xs:string"/>
    <xs:element name="workweek" type="xs:string"/>
    <xs:element name="salary" type="decimalType"/>
    <xs:element name="employmentType" type="xs:string"/>
    <xs:element name="positionInCompany" type="positionType"/>
  </xs:sequence>
</xs:complexType>

```

Под-елементите departmentName, workweek, salary, employmentType и positionInCompany дефинират различни характеристики на служителя. Например, departmentName представлява името на департамента, в който работи служителят, workweek - работната седмица, salary - заплатата (от тип decimalType), employmentType - видът на заетостта (например, пълен работен ден, непълен работен ден) и positionInCompany - позицията на служителя в организацията.

```

<xs:complexType name="positionType">
  <xs:sequence>
    <xs:element name="positionName" type="xs:string"/>
    <xs:element name="startDate" type="dateType"/>
  </xs:sequence>
</xs:complexType>

```

Под-елементите positionName и startDate дефинират информация за позицията на служителя. positionName представлява името на позицията (например, "Мениджър по продажби"), а startDate представлява датата на начало на заемане на тази позиция (от тип dateType). Този positionType е използван в employeeDataType.

```

<xs:complexType name="managerType">
  <xs:attribute name="firstNameRef" type="xs:string" use="required"/>
  <xs:attribute name="lastNameRef" type="xs:string" use="required"/>
  <xs:attribute name="idRef" type="xs:string" use="required"/>
</xs:complexType>

```

Атрибутите firstNameRef, lastNameRef и idRef представляват информация за мениджъра. firstNameRef и lastNameRef са атрибути за първо и последно име на мениджъра, а idRef представлява идентификационен номер на мениджъра. Всички тези атрибути са от тип xs:string и са маркирани като required, което означава, че те са задължителни и трябва да бъдат предоставени за всеки елемент от тип managerType.

```

<!--
<!DOCTYPE catalogue [
  <!ELEMENT catalogue (employeesList, departmentsList)>
  <!ELEMENT employeesList (employee+)>
  <!ELEMENT employee (personData, employeeData)>
  <!ELEMENT personData (firstName, middleName, lastName, id, birthday, telephoneNumber, residenceData)>
  <!ELEMENT residenceData (country, city, zipCode, address)>
  <!ELEMENT employeeData (departmentName, workweek, salary, employmentType, positionInCompany)>
  <!ELEMENT positionInCompany (positionName, startDate)>
  <!ELEMENT departmentsList (department+)>
  <!ELEMENT department (departmentName, manager)>
  <!ELEMENT manager EMPTY>
  <!ATTLIST manager
    firstNameRef CDATA #REQUIRED
    lastNameRef CDATA #REQUIRED
    idRef CDATA #REQUIRED
  >
  <!ELEMENT departmentName (#PCDATA)>
  <!ELEMENT positionName (#PCDATA)>
  <!ELEMENT startDate (#PCDATA)>
  <!ELEMENT country (#PCDATA)>
  <!ELEMENT city (#PCDATA)>
  <!ELEMENT zipCode (#PCDATA)>
  <!ELEMENT address (#PCDATA)>
  <!ELEMENT workweek (#PCDATA)>
  <!ELEMENT salary (#PCDATA)>
  <!ELEMENT employmentType (#PCDATA)>
  <!ELEMENT telephoneNumber (#PCDATA)>
  <!ELEMENT birthday (#PCDATA)>
  <!ELEMENT id (#PCDATA)>
  <!ELEMENT lastName (#PCDATA)>
  <!ELEMENT middleName (#PCDATA)>
  <!ELEMENT firstName (#PCDATA)>
]>
-->

```

Кода, който виждате, е XML DTD (Document Type Definition) декларация. Това е начин да се опише структурата и типовете данни, които могат да бъдат използвани в XML документ. В този конкретен случай, каталогът съдържа списъци от служители (**employeesList**) и отделения (**departmentsList**), като всяко отделение има свой списък със служители.

Декларациите са структурирани така:

1. **<catalogue>** е коренов елемент, който съдържа два под-елемента: **employeesList** и **departmentsList**.
2. **<employeesList>** съдържа списък с елементи **employee**.
3. **<employee>** се състои от два под-елемента: **personData** и **employeeData**.
4. **<personData>** съдържа информация за човека, като име, адрес, телефон и др.
5. **<residenceData>** е част от **personData** и съдържа информация за адреса на човека.
6. **<employeeData>** съдържа информация за служителя, като департамент, работна седмица, заплата и т.н.

7. **<positionInCompany>** е част от **employeeData** и съдържа информация за позицията в компанията, включително началната дата.
8. **<departmentsList>** съдържа списък с елементи **department**.
9. **<department>** съдържа информация за отделението и има един под-елемент **manager**, който е празен (**EMPTY**), но има атрибути за свързване с информация за мениджъра (име, фамилия, идентификационен номер).

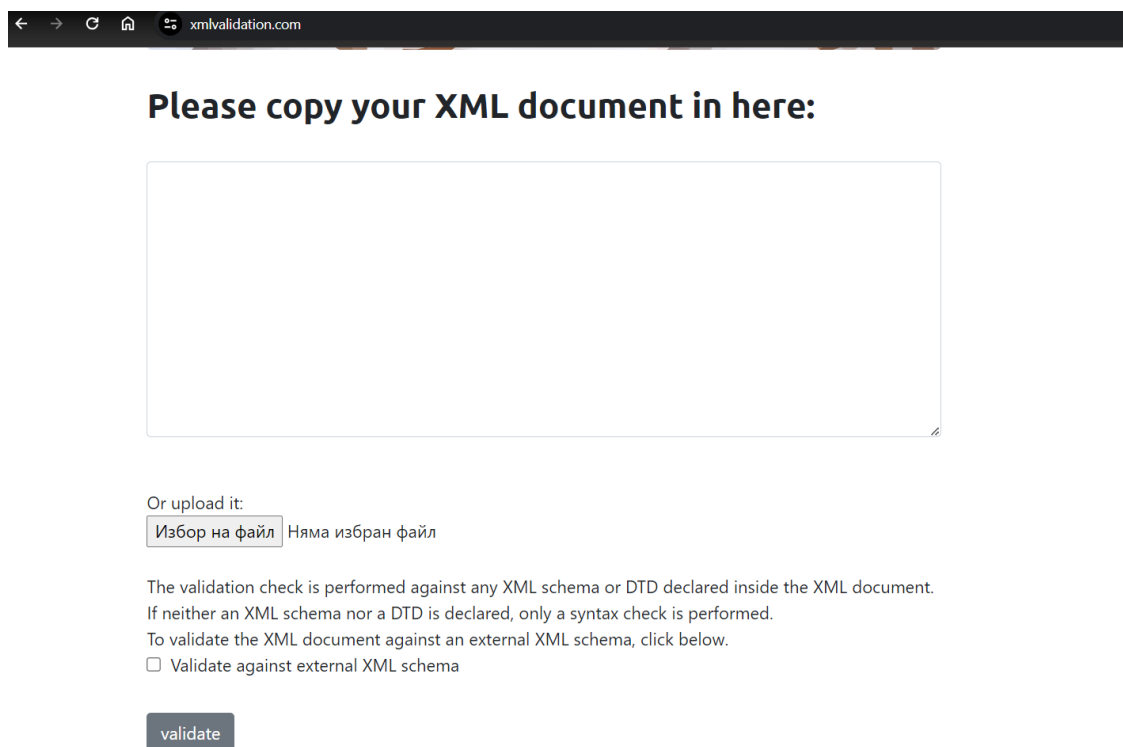
Текстовите елементи (**#PCDATA**) като **<departmentName>**, **<positionName>**, и други, съдържат текстова информация.

Този DTD дефинира структурата на XML документа, който ще използва този модел. Когато се създаде XML документ, той трябва да следва този формат, за да бъде валиден спрямо този DTD.

DTD файлът е добавен, като допълнителна валидация под формата на коментар, за да се избегне конфликт при валидирането с `schema.xsd`. DTD сегментът не пречи по никакъв начин да се визуализира `data.xml`.

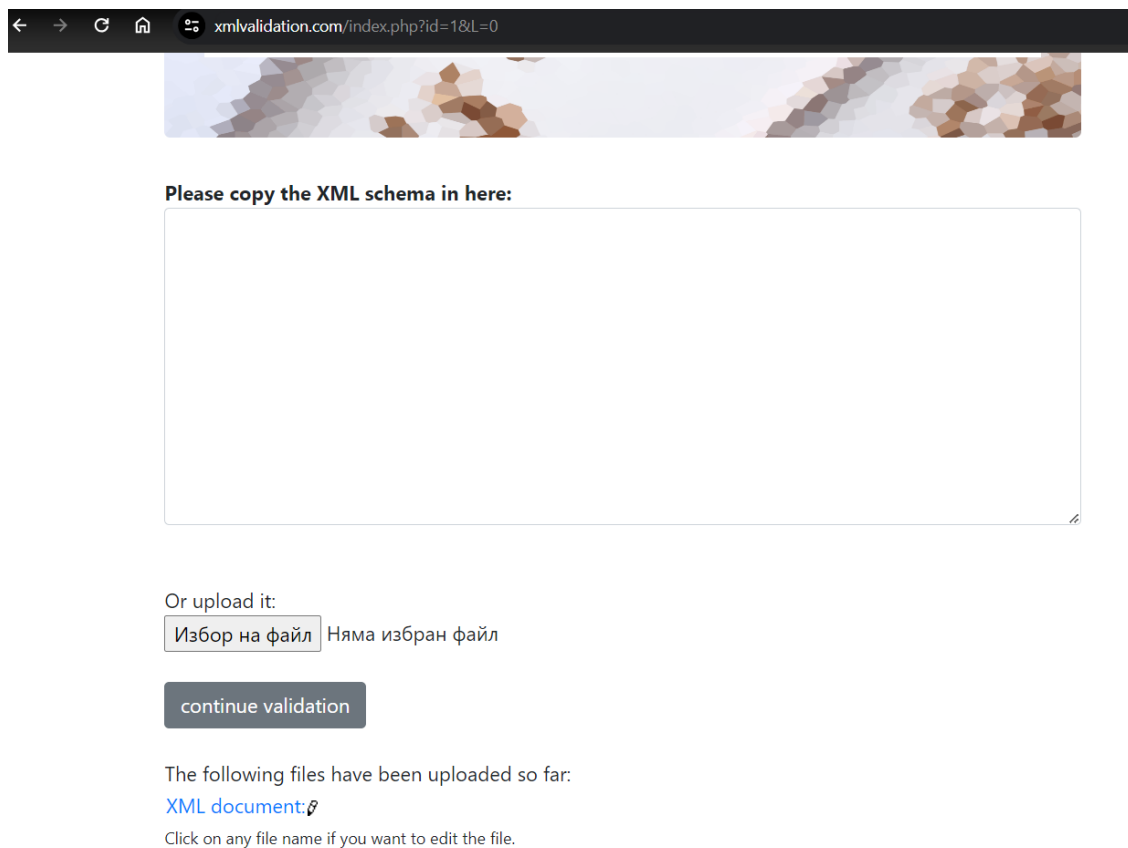
## 4 Тестване

За валидацията на XML документа спрямо външна XML Schema документ е използвана онлайн средата, намираща се на следната онлайн страница <https://www.xmlvalidation.com/>:

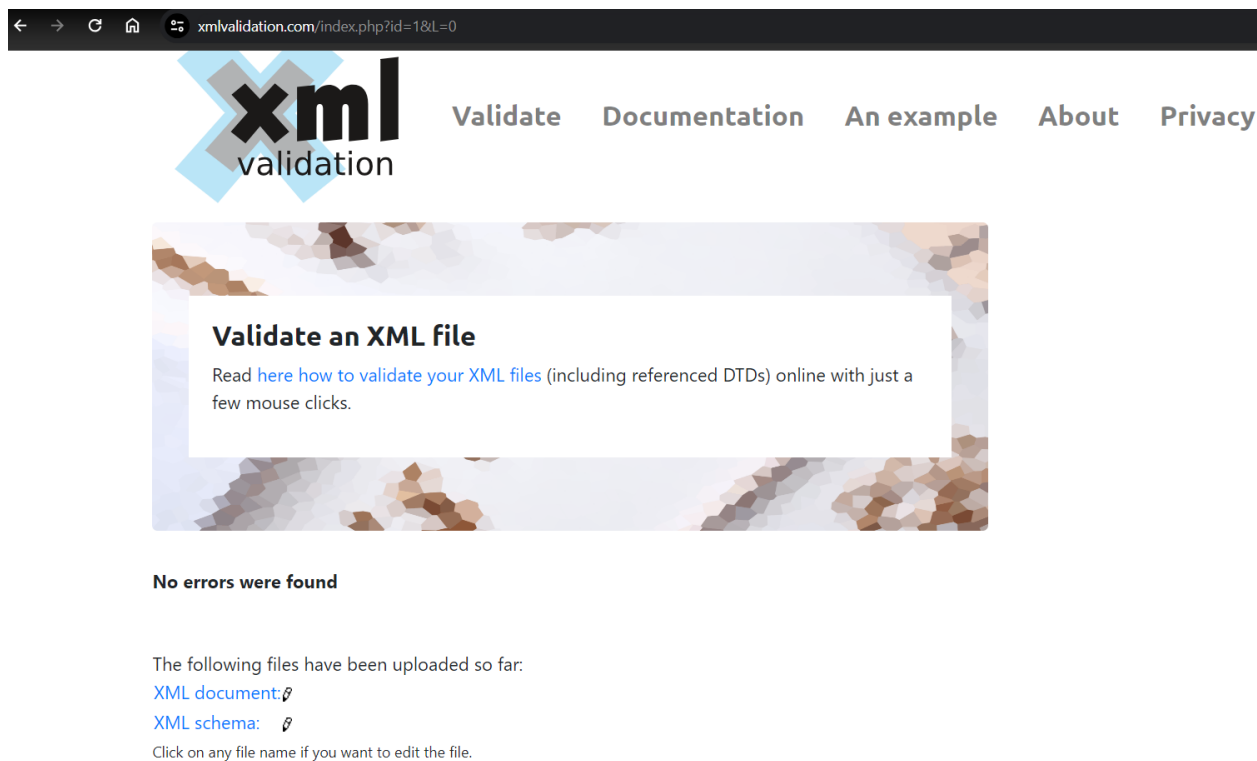


The screenshot shows a web browser window with the address bar displaying 'xmlvalidation.com'. The main heading reads 'Please copy your XML document in here:'. Below this is a large, empty rectangular text area for pasting XML code. Underneath the text area, it says 'Or upload it:' followed by a file selection button labeled 'Избор на файл' and the text 'Няма избран файл'. Below this, there is explanatory text: 'The validation check is performed against any XML schema or DTD declared inside the XML document. If neither an XML schema nor a DTD is declared, only a syntax check is performed. To validate the XML document against an external XML schema, click below.' This is followed by a checkbox labeled 'Validate against external XML schema', which is currently unchecked. At the bottom of the form is a dark grey button labeled 'validate'.

Фиг. 2. – Онлайн среда за валидиране – Избор на XML документ



Фиг. 3. – Онлайн среда за валидиране – Избор на XML Schema документ



Фиг. 4. – Онлайн среда за валидиране – Резултат след валидацията



Използван е IE (Internet Explorer) за графично генериране на съдържанието на XML документа. Резултатът от изпълнението на генерирането на кода е показано в следващите две фигури (Фиг. 5 и 6):

Image	First Name	Middle Name	Last Name	ID	Birthday	Telephone Number	Country	City	Zip Code	Address	Department Name	Workweek	Salary	Employment Type	Position Name	Start Date
	Boril	Georgiev	Stoyanov	14214214	1999-10-05	+359473829109	Bulgaria	Sofia	1001	3 Yoan Ekzarh Street	Backend	fullTime	2500.00	permanent	Junior Developer	2023-05-24
	Borislav	Ivanov	Georgiev	14214215	1989-05-03	+359473829110	Bulgaria	Sofia	1019	4 Kliment Ohridski Street	Frontend	partTime	3000.00	intern	Junior Developer	2023-12-08
	Stoyan	Ivanov	Iliev	14214216	1975-02-06	+359473829111	Bulgaria	Sofia	1008	7 Vasil Levski Street	Frontend	fullTime	6500.00	shift	Middle Developer	2015-06-01
	Ivan	Plamenov	Petrov	14214217	2001-10-05	+359473829131	Bulgaria	Sofia	1005	10 Hilar Petur Street	Backend	partTime	4000.00	intern	Trainee Developer	2021-08-08
	Plamena	Plamenova	Aleksandrova	14214218	2002-04-15	+35947382954	Bulgaria	Sofia	1005	24 Hristo Botev Street	Admin	fullTime	4300.00	permanent	Office assistant	2021-12-10

**Фиг. 5. – Резултат след графичното генериране на съдържанието на XML документа в HTML браузера**

Department Name	Admin First Name	Admin Last Name	Admin Image
Frontend	Kostadin	Dimitrov	
Backend	Dimitar	Yavorov	
Admin	Plamena	Aleksandrova	

**Фиг. 6. – Резултат след графичното генериране на съдържанието на XML документа в HTML браузера**

## 5 Заключение и възможно бъдещо развитие

Проектът изпълнява наложените изисквания успешно като каталогът, който е XML документ, генериран от XSL, представлява лесен шаблон за оформянето на каталози от всякакъв тип. Причината за това се дължи на функционалността да сортира по различни критерии служителите в дадена фирма. Проектът може да се разшири по следните няколко начина:

- Проектът обаче може да бъде оформен, като се разширят функционалностите като по-голям набор от данни за служители като образование, предходни кариерни позиции, препоръки от други компании, специализации, командировки в чужбина и прочие. Може да се добави акцент върху дейността на даден служител в дадената фирма. Може да се имплементира и база от данни, която да следи за служител на месеца и допълнително да се създаде сортировка на такава основа.
- Съществува вариант да се добави бутон, който при натискането си да генерира графично наново съдържанието на XML документа, за да може да се проследи по-лесно между новата страница и предишната относно служителите във фирмата. Подобна имплементация би ни отнела малко повече време за реализирането ѝ и поради това го отхвърлихме във финалната версия на проекта. Допълнително можем да добавим бутон, който да добавя служители и да се генерират тези данни към XML документа.
- Друг вариант за подобрене на проекта би бил да се създаде 3D галерия от профилните снимки на служителите, като се поддържат бутони за придвижване наляво и надясно от текущ служителя и при натискане върху профилната снимка на служителя да се генерира информация съответстваща в XML документа. Това би направило естетически сайта по-приятен за потребителя, който разглежда каталога.
- Друг вариант за подобрене на сайта би било да се разбият на определени групи служителите във фирмата, така че да се разграничат по-лесно екипите и съответстващите им проекти, по които работят. За реализацията на такава функционалност следва да се разшири XML файловия документ с още служители и да се добавят елементи, които разграничават потребителите по групи/екипи. По този начин може да се дефинира сортировка, която да бъде вътрешна за даден екип или външна/отделна за цялата фирма.

Като за обобщение, проектът удовлетворява изискванията по заданието и улеснява търсенето на потребители в една фирма по каталог, като изпълнява сортировка по критерий и дава способ на потребителя да погледне служителите в даден фирмен отдел с удобния интерфейс, който предлага генерираният HTML уебсайт.

## 6 Разпределение на работата

Работата по самият проект е била разпределена по следния начин:

- Борил Игнатов работи по файловете `data.xml`, `stylesheet.xsl` и `style.css`.
- Гена Генова работи по файловете `data.xml`, `schema.xsd` и добави снимките в папката `images`.

Работата по самата документация е била разпределена по следния начин:

- Борил Игнатов написа точките 4, 5, 6 и 7, като Гена Генова прегледа всяка една от тях.
- Гена Генова написа точките 1, 2, 3 и 7 от документацията, като Борил Игнатов прегледа и редактира всяка една от тях.

## 7 Използвани литературни източници и Уеб сайтове

1. Лекции по XML технологии за семантичен web, зимен семестър 2023/2024
2. <http://managementhelp.org/humanresources/index.htm>
3. <https://www.xmlvalidation.com/>
4. <https://developer.mozilla.org/en-US/docs/Web/CSS/text-decoration>
5. [https://www.liquidtechnologies.com/Reference/XmlStudio/XsdEditorNotation\\_BuiltInXsdTypes.html](https://www.liquidtechnologies.com/Reference/XmlStudio/XsdEditorNotation_BuiltInXsdTypes.html)
6. <https://www.softwaretestinghelp.com/how-to-open-xml-file/>
7. [https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ms754529\(v=vs.85\)](https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ms754529(v=vs.85))
8. [https://www.w3schools.com/xml/xml\\_display.asp](https://www.w3schools.com/xml/xml_display.asp)
9. <https://www.javatpoint.com/xslt-xsl-sort-element>
10. <https://www.vecteezy.com/free-vector/silhouette-profile-picture>
11. <https://www.vectorstock.com/>
12. <https://www.istockphoto.com/photos/default-avatar>
13. <https://www.vecteezy.com/vector-art/1840612-picture-profile-icon-male-icon-human-or-people-sign-and-symbol-vector>
14. [https://www.pngitem.com/middle/hxRbRT\\_profile-icon-png-default-profile-picture-png-transparent/](https://www.pngitem.com/middle/hxRbRT_profile-icon-png-default-profile-picture-png-transparent/)
15. <https://www.shutterstock.com/es/image-illustration/male-default-placeholder-avatar-profile-gray-582509560>
16. <https://www.w3.org/Style/XSL/>
17. <https://www.w3.org/TR/xslt20/>
18. <https://www.w3.org/TR/xmlschema11-1/>

## 8 Апендикс

При необходимост можете да добавите и допълнителни секции под формата на апендикси.

*Заб.: документацията на проекта се предава само в електронен вид (в MS Word и PDF формат), чрез качването на архив с документа и всички останали файлове по проекта, в задание за предаване на курсов проект в страницата на курса в Moodle.*

