



Génie Informatique et Digitalisation

Rapport de projet XML :

Healthcare virus

Yasser El Omrani Sara Outhounjite Marwa kadada

I. Introduction

1.Contexte du projet :

Le sujet "healthcare virus" se concentre sur l'étude, la prévention, le traitement et la gestion des virus dans le contexte des soins de santé. Les virus sont des agents pathogènes microscopiques qui peuvent infecter divers organismes, y compris les humains. Dans le domaine de la santé, la compréhension des virus revêt une importance particulière en raison de leur capacité à causer des maladies, parfois graves.

2.Objectifs du projet :

Dans le cadre de ce projet, plusieurs objectifs ont été établis pour assurer la réussite de notre réalisation. En premier lieu, notre principal but était de concevoir un document XML qui respecte scrupuleusement les normes et les bonnes pratiques du langage XML, tout en offrant une représentation complète et structurée des divers virus.

Un autre objectif significatif était de maîtriser l'utilisation des DTDs afin de définir une grammaire XML cohérente et rigoureuse. Nous avons visé à garantir que notre document respecte toutes les contraintes syntaxiques et les règles établies, assurant ainsi sa conformité aux normes XML.

De manière concomitante, nous nous sommes fixés pour objectif d'exploiter les fonctionnalités avancées des DTDs pour enrichir la structure de notre document, permettant ainsi d'intégrer des informations de manière organisée et logique.

De plus, nous avons envisagé l'utilisation des XSLT comme un moyen de valoriser notre document XML en le transformant en un site web interactif et convivial. Cette transformation avait pour but non seulement de présenter le contenu du document de manière esthétique, mais aussi d'offrir une expérience utilisateur intuitive et agréable, en créant une interface web dynamique.

II. Conception de la grammaire XML :

1.Spécification des balises et de leur syntaxes :

Le code XML que vous avez fourni décrit une liste de virus avec des informations telles que leur nom, date de découverte, caractéristiques, symptômes, méthodes de prévention et histoire. Voici une spécification des balises utilisées dans ce code :

1. `<?xml version="1.0" encoding="UTF-8"?>` : Déclaration XML indiquant la version utilisée et l'encodage du document.

2. `<!DOCTYPE viruses SYSTEM "virus.dtd">` : Déclaration du type de document (DOCTYPE). Ici, elle fait référence à un fichier de définition de type de document (DTD) appelé "virus.dtd", bien que le fichier DTD lui-même ne soit pas inclus ici.

3. `<?xml-stylesheet type="text/xsl" href="virus.xslt" version="1.0"?>` : Indication d'une feuille de style XSLT pour la transformation du document XML. La feuille de style est définie dans un fichier externe "virus.xslt".

4. `<viruses>` : Balise racine qui englobe toutes les informations sur les virus.

5. `<virus>` : Balise représentant chaque entité virus avec ses informations associées.

- `<name>` : Balise indiquant le nom du virus.
- `<discoveryDate>` : Balise pour la date de découverte du virus.
- `<characteristics>` : Balise pour les caractéristiques du virus.
- `<symptoms>` : Balise pour les symptômes associés au virus.
- `<prevention>` : Balise pour les méthodes de prévention du virus.
- `<history>` : Balise pour l'histoire ou les antécédents du virus.

Le code se termine par la fermeture des balises correspondantes (`</virus>`, `</viruses>`). Chaque `<virus>` représente une entrée distincte dans la liste des virus. Le code suit une structure XML bien formée et hiérarchique, permettant une représentation organisée et lisible des informations sur différents virus.

2.Utilisation des DTDs :

Les TDTD (Type Definitions in Document Type Definitions) que nous avons fournis spécifient la structure autorisée pour le document XML.

Ils définissent les éléments autorisés, les sous-éléments, et le type de contenu autorisé dans ces éléments.

Voici une explication de chaque ligne dans vos DTD par rapport au code XML :

`<!ELEMENT viruses (virus+)>` : Cette déclaration indique que l'élément racine "viruses" doit contenir un ou plusieurs éléments "virus". L'opérateur "+" signifie "une ou plusieurs occurrences".

`<!ELEMENT virus (name, discoveryDate, characteristics, symptoms, prevention, history)>` : Cette déclaration définit la structure de l'élément "virus". Un élément "virus" doit contenir les sous-éléments suivants dans cet ordre : "name", "discoveryDate", "characteristics", "symptoms", "prevention", et "history".

`<!ELEMENT name (#PCDATA)>` : Cette déclaration indique que l'élément "name" ne peut contenir que du texte brut (Parsed Character Data - PCDATA). Il ne peut pas contenir d'autres éléments XML.

<!ELEMENT discoveryDate (#PCDATA)> : De même, cela spécifie que l'élément "discoveryDate" ne peut contenir que du texte brut.

<!ELEMENT characteristics (#PCDATA)> : La même logique s'applique à "characteristics".

<!ELEMENT symptoms (#PCDATA)> : Cela spécifie que l'élément "symptoms" ne peut contenir que du texte brut.

<!ELEMENT prevention (#PCDATA)> : De même, cela s'applique à "prevention".

<!ELEMENT history (#PCDATA)> : Cela spécifie que l'élément "history" ne peut contenir que du texte brut.

III. Validation syntaxique :

Afin de garantir que notre document XML soit conforme aux normes établies, nous avons employé l'environnement de développement intégré (IDE) Visual Studio. Grâce à cet outil puissant, nous avons importé notre fichier XML ainsi que sa DTD associée. En utilisant les fonctionnalités de validation intégrées de Visual Studio, nous avons initié le processus de vérification du document XML par rapport à la DTD que nous avons élaborée.

Visual Studio a détecté les erreurs et les incohérences potentielles dans notre document XML en se basant sur les règles définies dans notre grammaire XML. Nous avons examiné attentivement ces avertissements et erreurs, nous permettant de localiser précisément les parties du document nécessitant des ajustements. En effectuant les corrections nécessaires, telles que la modification de balises ou d'attributs, nous avons assuré la conformité de notre document à notre DTD.

L'utilisation de Visual Studio nous a fourni un environnement convivial et efficace pour effectuer ces vérifications, garantissant ainsi l'intégrité de notre document XML. Cette étape cruciale a assuré que notre document respecte les spécifications que nous avons définies, le rendant prêt à être utilisé dans divers contextes et applications.

IV. Création du site web :

1. Processus de transformation du document XML en site web :

Le processus de transformation d'un document XML en un site web comprend plusieurs étapes :

1. Analyse du document XML : Cette étape prend en compte le traitement de toutes les balises, les attributs, les noms d'éléments, etc. C'est à ce stade précis que le document est analysé pour déterminer sa structure.

2. Conversion du XML en HTML : La deuxième étape consiste à convertir le fichier XML en HTML (HyperText Markup Language) pour qu'il soit compatible avec les navigateurs Web. Cela implique d'utiliser des outils spécifiques pour convertir le document XML en code HTML.

3. Création de feuilles de style CSS : Une fois que le document XML a été converti en HTML, il est temps de créer les feuilles de style CSS (Cascading Style Sheets). Cela permet de donner au site web un look professionnel, ainsi qu'une structure facile à lire.

4. Génération du contenu dynamique : C'est l'étape où le contenu dynamique est généré. Les outils utilisés à ce stade dépendent des besoins et des exigences spécifiques du site web.

5. Intégration de codes supplémentaires : À cette étape, les codes supplémentaires sont intégrés au site web. Il peut s'agir de scripts de fonctionnalités spécifiques qui doivent être intégrés pour que le site Web fonctionne correctement.

6. Intégration des images et des médias : Dans cette étape, les images et les médias sont ajoutés au site web.

7. Tests et validation : Enfin, avant la mise en production, le site Web doit être testé et validé pour détecter les erreurs et les éventuels problèmes. Cela permet de s'assurer que le site web est prêt pour une utilisation en ligne.

2.Présentation du site web :



Smallpox

Discovery Date: 18th century

Characteristics: Respiratory transmission, fever, rash

Symptoms: Fever, rash, body aches

Prevention: Vaccination, isolation of infected individuals

History: The origin of smallpox is unknown. The finding of smallpox-like rashes on Egyptian mummies suggests that smallpox has existed for at least 3,000 years. The earliest written description of a disease like smallpox appeared in China in the 4th century CE (Common Era).

Polio

Discovery Date: 1789

Characteristics: Fecal-oral transmission, paralysis

Symptoms: Paralysis, fever, headache

Prevention: Vaccination, hand hygiene

History: Polio has existed since ancient times and was documented in the written record in the 18th and 19th centuries. While it affected children around the world for millennia, the first known clinical description of polio, by British doctor Michael Underwood, was not until 1789, and it was formally recognized as a condition in 1840 by German physician Jakob Heine.