Université M'Hamed Bougara de Boumerdès
Faculté des Sciences
Département d'Informatique

Application Web de Vente de Bijoux en Ligne basée sur XML

Participants :
- MEDJAHED Ikram
- HAMCHAOUI Badreddine

Groupe : TI 03

CAWA (M1 - S2) – Année universitaire 2024/2025

Présentation Générale

Ce projet web est une boutique en ligne de bijoux (bagues et boucles d'oreilles) mettant en avant l'usage de XML comme format central de données. L'application permet de naviguer, rechercher, filtrer et commander des bijoux

Architecture du Projet

Modélisation des Données (XML + XSD)

- Le fichier `bijoux.xml` contient une liste de bijoux structuré (catégories : bagues, boucles d'oreilles), et des commandes.
- La structure est validée par le schéma `bijoux.xsd` qui impose des contraintes sur les types et les valeurs autorisées.

Base de Données PostgreSQL

- Une table `table_xml` a été créée avec un champ de type XML contenant l'intégralité du fichier `bijoux.xml`
- Des requêtes XPath sont utilisées via PostgreSQL pour extraire dynamiquement les bijoux selon certains critères (prix, catégorie...).

Transformation XSLT (XML → HTML)

- Le fichier `bijoux.xslt` permet de transformer les données **XML** en une page **HTML** statique.
- La transformation XSLT se fait côté client via JavaScript, à l'aide de `XSLTProcessor`.

Backend – API REST (Python + Flask)

L'API expose les endpoints suivants :

- `GET /bijoux` :retourne la liste complète des bijoux au format XML, avec possibilité de filtrer par mot-clé (search) et de trier les résultats (sort) via des paramètres optionnels.
- `GET /bijoux/categorie/<categorie>` : filtre les bijoux par catégorie en utilisant XPath.
- `GET /bijoux/<nom>` : récupère les bijoux par nom.

- `GET /xml` : renvoie le contenu brut du fichier XML contenant les données des bijoux.
- `POST /order` : ajout d'une commande au format XML dans la base.

Toutes les interactions **REST** utilisent **XML** comme format de données en entrée/sortie, avec filtrage et recherche via **XPath**.

Frontend – Interface Dynamique

Une interface dynamique permet de :

- La transformation XSLT se fait côté client via JavaScript, à l'aide de `bijoux.xslt` et `bijou.xslt`.
- Filtrer les bijoux (par catégorie), effectuer des recherches et trier selon le prix, grâce à des appels à l'API **REST**.
- Manipulation dynamique du **DOM** et requêtes `fetch`.

Choix techniques

- **XML/XSD/XSLT** pour la structuration, la validation, et la transformation des données.
- **Flask (Python)** pour la gestion des routes et des services **REST**.
- **PostgreSQL** comme base de données, exploitant un champ XML natif.
- **React + Next.js** pour le front-end dynamique, l'interaction DOM et la transformation XSLT.
- **XPath** pour les recherches, le filtrage et la gestion des commandes.
- **Tailwind CSS** pour le stylage rapide et réactif du front-end.
- **Mise en cache** des requêtes XML pour améliorer les performances.

Lien vers la vidéo explicative et le code source

Lien vers la vidéo

Lien vers le code

Note : L'explication audio a été réalisée à l'aide d'une intelligence artificielle, en raison de la non-disponibilité d'un microphone de bonne qualité.