**Inchiriere Biciclete**

**Autor: Ristoiu George-Aurel, 332AA**

**Cuprins:**

1. **Descrierea domeniului**
2. **Proiectare XML**
3. **Proiectare DTD**
4. **Proiectare XSD**
5. **Descrierea aplicatiei**
6. **Bibliografie**
7. Descrierea domeniului

Acest domeniu se bazeaza pe faptul ca utilizatorul are nevoie sa introduca:

- datele personale pentru eventualitatea in care bicicleta nu este returnata

- informatiile despre biciclete (tip, numar)

- numarul de ore aproximativ pentru care vrea sa inchirieze

1. Proiectare XML

Pentru proiectarea fisierului xml am folosit o ramura de root “rentals”, care va contine toate inchirierile de biciclete cu tagul “rental”

Pentru fiecare inchiriere de tip “rental” avem informatiile despre client “clientInformation” si despre inchiriere “rentalDetails”.

Ramura de tip “clientInformation” contine:

* Nume: “name”
* CNP: “SSN” (Social Security Number)
* Seria si numarul de buletin: “securitySeries” si “securityNumber”
* Numar telefon: “phoneNumber”

Ramura de tip “rentalDetails” contine:

* Data: “date”
* Numarul de ore: “numberOfHours”
* Detaliile bicicletelor: “bikeDetails”

Ramura “bikeDetails” contine la randul sau tipu de Bicicleta si numarul de ore(“bikeType”, “numberOfBikes”)

1. Proiectare DTD

Putem avea 0 sau mai multe inchirieri de biciclete

<!ELEMENT rentals (rental+)>

<!ELEMENT rental (clientInformation, rentalDetails)>

Avem un singur atribut si anume id-ul fiecarei inchirieri

<!ATTLIST rental id CDATA #REQUIRED>

Elementul “client information” ce contine campurile: nume, CNP, serie si numar de bulletin si numarul de telefon

<!ELEMENT clientInformation (name, SSN, securitySeries, securityNumber, phoneNumber)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT SSN (#PCDATA)>

<!ELEMENT securitySeries (#PCDATA)>

<!ELEMENT securityNumber (#PCDATA)>

<!ELEMENT phoneNumber (#PCDATA)>

<!ELEMENT rentalDetails (date, numberOfHours, bikeDetails)>

Elementul “rentalDetails” ce contine campurile: data, numar de ore si detailii despre biciclete

<!ELEMENT date (#PCDATA)>

<!ELEMENT numberOfHours (#PCDATA)>

<!ELEMENT bikeDetails (bykeType, numberOfBykes)>

Si campul “bikeDetails” ce contine tipul de Bicicleta si numarul de biciclete

<!ELEMENT bykeType (#PCDATA)>

<!ELEMENT numberOfBykes (#PCDATA)>

1. Proiectare XSD

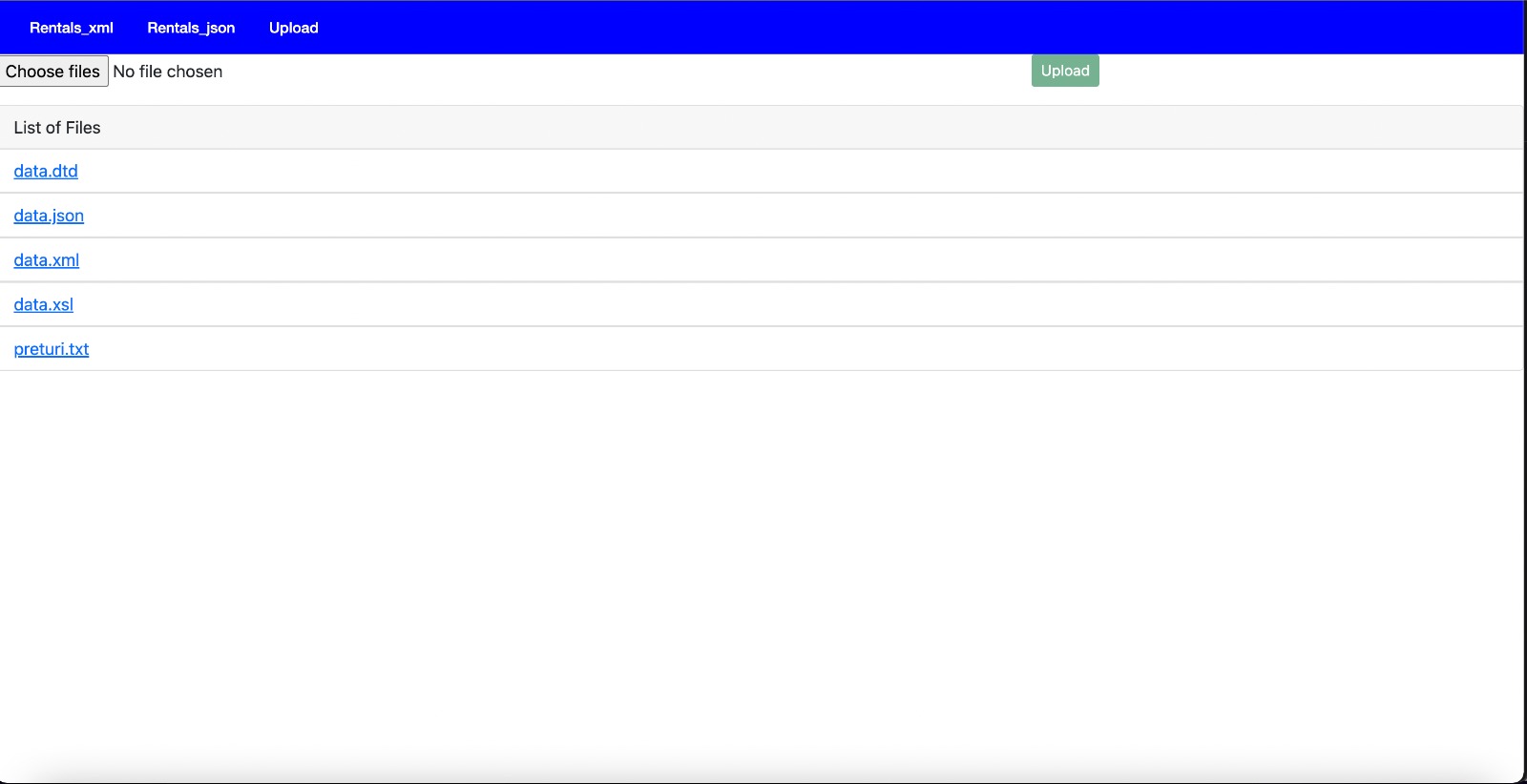
Restrictii propuse:

* Numarul maxim de elemenete “rental” nu este specificat
* Data este de tip date si apare o singura data
* Restul elementelor sunt fie elemente compuse, fie stringuri ce apar o singura data

1. Descrierea aplicatiei

Aplicatia contine 3 rute:

* Pagina unde este parsat fisierul xml
* Pagina unde este parsat fisierul json
* Pagina unde sunt incarcate fisierele xml si json

 Pe langa cele 2 fisiere, trebuie incarcate si fisierele xsl si dtd deoarece fara acestea, nu vom putea vizualiza rezultatul aplicarii foii de stiluri asupra fisierului xml. Acestea vor fi incarcate initial si vor aparea in lista de fisiere incarcate

Se va apasa butonul “Choose files”, se vor selecta fisierele xml si dtd (pot fi gasite in folderul “./backendXML/db” sau pot fi folosite fisierele din arhiva cu conditia de a fi redenumite cu “data.xml” si “data.json”) si se va apasa pe butonul de upload. In momentul asta, lista va contine si cele 2 fisiere incarcate. Daca inca nu apar elementele in tabele in celelalte 2 rute, se da refresh la pagina. Pentru a deschide fisierul de view cu xml si xsl se apasa pe numele fisierului din lista de pe pagina de upload. La fel si pentru celelalte daca se doreste vizualizarea continutului fisierelor.

1. Bibliografie
2. <https://stackoverflow.com/questions/10904448/node-to-parse-xml-using-xml2js>
3. <https://nodejs.dev/learn/reading-files-with-nodejs>
4. <https://www.geeksforgeeks.org/file-uploading-in-node-js/>