CST3130 Advanced Web Development with Big Data

Coursework 1 Project Report

Jan Ciepiela M00779169



CST3130 Advanced Web Development with Big Data	1
Project Overview	3
Websites scraped	3
Data flow	4
Database Diagram	6
Unit testing	7

Project Overview

For the first piece of coursework I have created a project that shows the flow of data from being scraped from the website, to being put through to a database, to being displayed in a web application. The project consist of a few components working together. Java program is used to scrape multiple websites using JSoup to extract content from them, working on five threads, and place it into a SQL database with the help of Hibernate. The Java project is built using Maven, which also allows for executing JUnit tests created for the program. Another part is implemented using Javascript. Web service is used to retrieve data from the database and parse it to JSON, which is then retrieved by Axios and displayed on the web application using Vue.js. The dependencies and web service are managed with the help of Node.js and Express.js.

Websites scraped

The list of websites used for the purpose of the project:

• Gearbest: https://www.gearbest.com/

• Ritzcamera: https://ritzcamera.com/

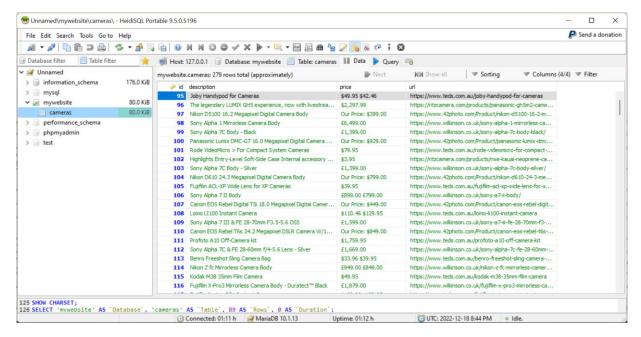
• Teds: https://www.teds.com.au/

Wilkinson: https://www.wilkinson.co.uk/

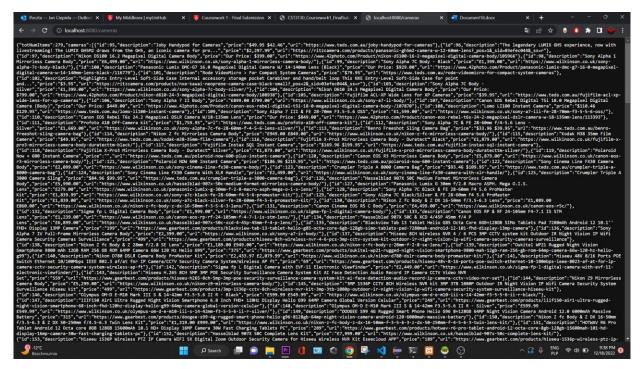
42photo: https://www.42photo.com/

Data flow

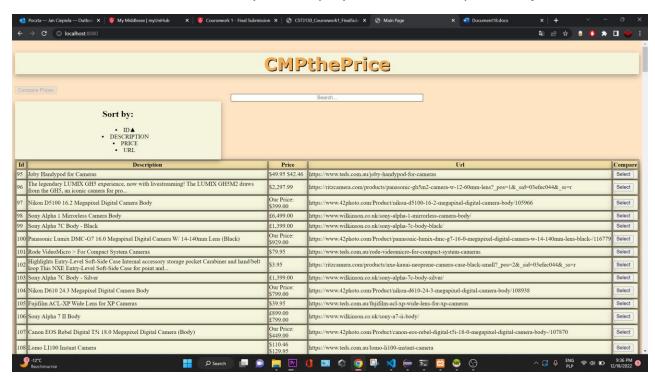
Once the data has been scraped, Hibernate is used to place it into a database. The view from HeidiSQL with the data placed in a "cameras" table:



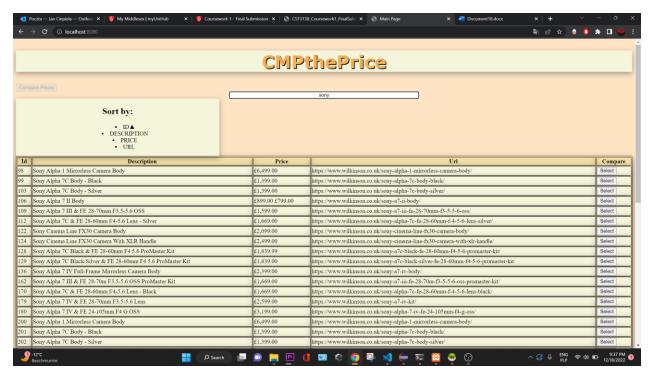
The data is then retrieved by the web service and parsed into JSON. An example of the contents parsed after using the path "/cameras":



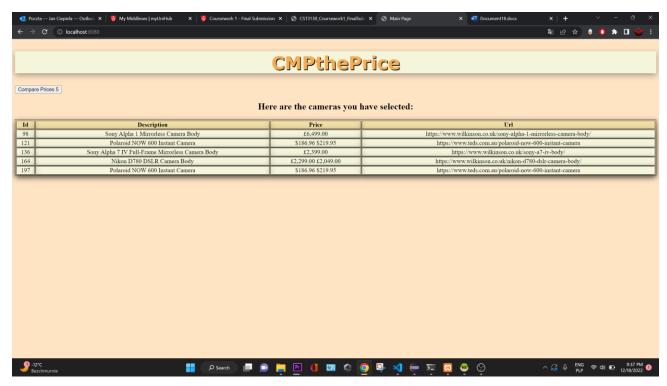
Data is then loaded into an array and displayed with the help of Vue.js:



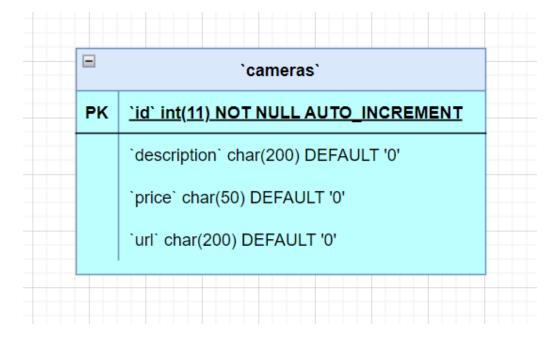
User can search the products, sort them by id, description, price and url as well as increasingly or descresingly:



Upon selection of at least two products, user can then compare them on the comparison page, to which one may switch using the button in the top left corner. The view from the comparison page:



Database Diagram



Unit testing

To verify if the project works correctly there have been designed five JUnit tests to examine the Java program, specifically the scrapers:

As well as two Javascript tests using Mocha and Chai to examine the connection to the database and the default API route:

```
PS D:\www> npm test

> www@1.0.0 test
> mocha --timeout 10000

App listening on port 8080

First Test Collection
    ✓ should test default API route
    ✓ Should connect to database without an error

2 passing (44ms)
```