University of the West of England
BRISTOL

MODULAR PROGRAMME

COURSEWORK ASSESSMENT SPECIFICATION

Module Details

Module Code	Run	Module Title
UFCFR5-15-3	21JAN/1	ADVANCED TOPICS IN WEB DEVELOPMENT 2
Module Leader	**Module Coordinator**	**Module Tutors**
Prakash Chatterjee		P Chatterjee
Component and Element Number		**Weighting: (% of the Module's assessment)**
B: CW1		50%
Element Description		**Total Assignment time**
BUILD, TEST AND DOCUMENT A WEB-BASED APP.		36 hours

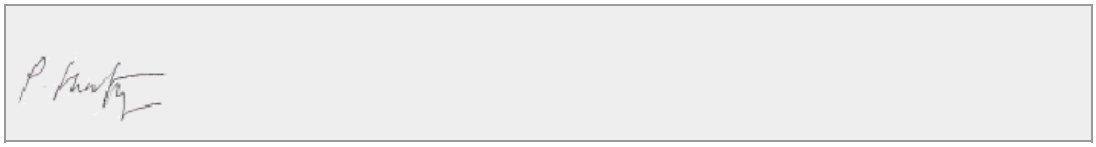
Dates

Date Issued to Students	Date to be Returned to Students
22 Feb 2021	4 working weeks after hand-in.
Submission Place	**Submission Date**
Blackboard	22 April 2021
	Submission Time
	2.00 pm

Deliverables

Blackboard submission with attached report in MARKDOWN format.

Module Leader Signature



UFCFR5-15-3: Advanced Topics in Web Development 2

Assignment Specification 2020-21

Learning Goals & Outcomes

- Learn to model, clense & normalize substantial real-world big data (188 mb+);
- Understand the data cleansing and normalization processes by writing PHP scripts to process and convert the data to first (cleansed) CSV and then (normalized) XML;
- Gain knowledge and skill in the design, implementation & visualisation of data using web based charting and mapping API's.
- Fulfil the submission requirement of providing the two conversion scipts for benchmarking and one XSD schema file for validation purposes.
- Extend your skills in the use use of key technologies: PHP, XML & XPATH, Parsing, DOM, JavaScript & JSON.
- Learn and use the MARKDOWN markup syntax.

Context: Measuring Air Quality

Levels of various air borne pollutants such as Nitrogen Monoxide (NO), Nitrogen Dioxide (NO_2) and particulate matter (also called particle pollution) are all major contributors to the measure of overall air quality.

For instance, NO_2 is measured using micrograms in each cubic metre of air (µg m^3). A microgram (µg) is one millionth of a gram. A concentration of 1 µg m^3 means that one cubic metre of air contains one microgram of pollutant.

To protect our health, the UK Government sets two air quality objectives for NO_2 in their Air Quality Strategy

1. The hourly objective, which is the concentration of NO_2 in the air, averaged over a period of one hour.
2. The annual objective, which is the concentration of NO_2 in the air, averaged over a period of a year.

The following table shows the colour encoding and the levels for Objective 1 above, the mean hourly ratio, adopted in the UK.

Index	1	2	3	4	5	6	7	8	9	10
Band	Low	Low	Low	Moderate	Moderate	Moderate	High	High	High	Very High
µg/m³	0-67	68-134	135-200	201-267	268-334	335-400	401-467	468-534	535-600	601 or more

Further details of colour encodings and health warnings can be found at the DEFRA Site.

The Input Data

The following ZIP file provides data ranging from 2004 to 31 Dec 2019 taken from 16 monitoring stations in and around Bristol.

Monitors come and go and may suffer down times, so the data isn't complete for all stations at all times.

Download & save the data file `air-quality-data-2004-2019.zip`

Shown here is the first 7 lines of the file:

```
Date Time;NOx;NO2;NO;SiteID;PM10;NVPM10;VPM10;NVPM2.5;PM2.5;VPM2.5;CO;O3;SO2;Temperature;RH;Air Pressure;Location;geo_point_2d;DateStart;DateEnd;Current
2019-01-01T00:00:00+00:00;30.71;25.69;3.27;500;13.53;;;;;;;;;;;;Temple Way;51.4579497129,-2.58398909033;2017-08-01T00:00:00+00:00;;True
2017-06-23T21:00:00+00:00;55.75;23.5;21.25;270;;;;;;;;;;;;Wells Road;51.4278638883,-2.56374153315;2003-05-23T00:00:00+00:00;;True
2017-06-23T23:00:00+00:00;42.0;25.0;11.0;463;;;;;;;;;;;;Fishponds Road;51.4780449714,-2.53523027459;2009-03-13T00:00:00+00:00;;True
2017-06-24T00:00:00+00:00;31.75;27.25;3.25;215;;;;;;;;;;;;Parson Street School;51.432675707,-2.60495665673;2002-02-01T00:00:00+00:00;;True
2017-06-24T00:00:00+00:00;42.25;16.25;16.5;270;;;;;;;;;;;;Wells Road;51.4278638883,-2.56374153315;2003-05-23T00:00:00+00:00;;True
```

Note the following:

There are 18 stations (monitors):

188 => 'AURN Bristol Centre',
203 => 'Brislington Depot',
206 => 'Rupert Street',
209 => 'IKEA M32',
213 => 'Old Market',
215 => 'Parson Street School',
228 => 'Temple Meads Station',
270 => 'Wells Road',
271 => 'Trailer Portway P&R',
375 => 'Newfoundland Road Police Station',
395 => "Shiner's Garage",
452 => 'AURN St Pauls',
447 => 'Bath Road',
459 => 'Cheltenham Road \ Station Road',
463 => 'Fishponds Road',
481 => 'CREATE Centre Roof',
500 => 'Temple Way',
501 => 'Colston Avenue'

Each line represents one reading from a specific detector. Detectors take one reading every hour. If you examine the file using a programming editor, Notepad++ can handle the job, you can see that the first row gives headers and there are another 1344953287 (1.3 million+) rows (lines). There are 22 data items (columns) per line.

The schema is given below:

measure	desc	unit
Date Time	Date and time of measurement	datetime
NOx	Concentration of oxides of nitrogen	µg/m3
NO2	Concentration of nitrogen dioxide	µg/m3
NO	Concentration of nitric oxide	µg/m3
SiteID	Site ID for the station	integer
PM10	Concentration of particulate matter <10 micron diameter	µg/m3
NVPM10	Concentration of non - volatile particulate matter <10 micron diameter	µg/m3
VPM10	Concentration of volatile particulate matter <10 micron diameter	µg/m3
NVPM2.5	Concentration of non volatile particulate matter <2.5 micron diameter	µg/m3
PM2.5	Concentration of particulate matter <2.5 micron diameter	µg/m3
VPM2.5	Concentration of volatile particulate matter <2.5 micron diameter	µg/m3
CO	Concentration of carbon monoxide	mg/m3
O3	Concentration of ozone	µg/m3
SO2	Concentration of sulphur dioxide	µg/m3
Temperature	Air temperature	°C
RH	Relative Humidity	%
Air Pressure	Air Pressure	mbar
Location	Text description of location	text
geo_point_2d	Latitude and longitude	geo point
DateStart	The date monitoring started	datetime
DateEnd	The date monitoring ended	datetime
Current	Is the monitor currently operating	text

Task 1: Cleanse and Refactor the Data (20 marks)

The original file is large and unwieldy, with some missing data rows and hence will be hard to process quickly and efficiently for different tasks.

Using a Divide & Conquer Strategy, we can break up the file into smaller more optimised data chunks. These will be much easier and more efficient to process.

Write a PHP script (`extract-to-csv.php`) to process the input file (`air-quality-data-2004-2019.csv`) and output 18 smaller `csv` files holding data for each monitoring station.

These output eighteen output files should be named as `data_188.csv` , `data_203.csv` , `data_206.csv` ... etc. with all empty data lines filtered out.

They should only hold 16 columns of essential data - the first 14 columns (pollution data) & column 18 + 19 (location & geo-location) with other columns filtered out.

You should also include only those records that have values for **either the NOx reading (column 2) or the CO reading (column 12)** i.e. if both are missing then exclude the record..

Each file will have the first line holding the following header:

```
siteID,ts,nox,no2,no,pm10,nvpm10,vpm10,nvpm2.5,pm2.5,vpm2.5,co,o3,so2,loc,lat,long
```

Notice the slight reorganization with `SiteID` moving from column 5 to 1 and Column 2 `ts` now holds a converted UNIX timestamp rather than a date/time string.

Submission File: A php sript file listing - ***extract-to-csv.php***

Note: If this file is unavailable after the submission date - a mark of 0 will be automatically assigned for this part.

Coding Tip: Make use of PHP's `fgets()` and `fputs()` functions. These are highly optimized libraries written in C and can easily handle the task. Avoid creating memory intensive intermediate arrays and other structures. Simply read a line and write it out to a specific station file. Avoid opening and closing lots of files repeatedly (very time consuming).

You will need to set the following PHP flags given the large size of the input file.

```php
# set timezone
@date_default_timezone_set("GMT");

ini_set('memory_limit', '512M');
ini_set('max_execution_time', '300');
ini_set('auto_detect_line_endings',TRUE);
```

Task 2: Data Transformation, Normalisation & XML Validation (20 marks)

2a. Write a PHP script - *`normalize-to-xml.php`* to transform each data CSV file to the following example XML structure (12 marks):

These files should be named `data_203.xml` , `data_206.xml` , `data_209.xml` ... etc. and formatted as follows (447 - Bath Road example):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<station id="447" name="Bath Road" geocode="51.4425372726,-2.57137536073">
    <rec ts="1287277200" nox="117.0" no="52.0" no2="64.75" />
    <rec ts="1287288000" nox="104.75" no="46.75" no2="57.25" />
    <rec ts="1287291600" nox="54.5" no="39.75" no2="14.5" />

    <!--
        .
        .
        Another  60000+ records -->
        .
        .
    -->
</station>
```

Notice this XML file is now normalized - not holding repeating data (station id, name & geocode) and only holding the attributes for each record which have values.

Submission File: A php sript file listing - ***normalize-to-xml.php***

This script should generate the required XML files to the current directory.

Note: If this file is unavailable after the submission date - a mark of 0 will be automatically assigned for this part.

Coding Tip: Again make use of PHP's `fgets()` or even `fgetcsv()` functions. Also the `XMLWriter()` library is useful here.

2b. Design and write a XSD 1.1 Schema file to validate the data XML files.(8 marks):

Design and implement a XML Schema (XSD) (`air-quality.xsd`) to validate the XML files generated above.

Marks will depend on the overall design, structure & strictness of the schema.

Submission File: A XSD 1.1 Schema file - ***air-quality.xsd***

Note: If this file is unavailable for reading after the submission date - a mark of 0 will be automatically assigned for this part.

Coding Tip: Make use of the oXygen XML editor in the Labs. This is a powerful tool and will reward learning.

Task 3: Charts Visualisation (20 marks)

Use Google Chart or any other charts API to visualise the following charts:

1. A scatter chart to show a years worth of data (averaged by month) from a specific station for Carbon Monoxide (NO) at a certain time of day - say 08.00 hours.
2. A line chart showing levels in any 24 hour period on any day (user selectable) for any of the six stations (user selectable) for any of the major pollutants (nox, no, no2) in the date range downloaded.

All charts should give an accurate rendition of the data set at an appropriate resolution.

Note that you can restrict the data to the range 01/01/2015 to 31/12/2019

Task 4: Maps Visualisation (20 marks)

Use Google Maps or any other mapping API to visualise the data:

You are encouraged to be creative and make use of the data in any way you decide.

Marks will be based on the creative use of the selected API, appropriate use of the data set & the usability and aesthetics of the user interface.

Note that you can restrict the data to the range 01/01/2015 to 31/12/2019

Task 5: Reflection & Report (20 marks)

A report in Markdown format to include:

- A discussion of parsing methods and tools and when to use streaming parsers over DOM parsers for document processing (up to 600 words) (8 marks);
- Links to all code and data files for examination and marking (8 marks);
- How you might go about refactoring and extending the charting and data visualisation functionality you have implemented.(4 marks)

A reflective discussion with comprensive links to working code / layouts will achieve the highest marks in this section.

This report must also link to all source code created (i.e. *.phps files for line by line inspection and NOT simply executables).

Marking Criteria and Scheme

Code Benchmark Tests will be carried out on a machine with an Intel i3-4130 CPU @ 3.40GHz with 8.00 GB of memory running Windows 10 and XAMPP.

Task 1: Cleanse and Refactor the Data (20 marks)

A benchmarking script will use the script ***extract-to-csv.php*** and run the following tests.

Benchmark Test 1a: Efficiency (Time)
How long the script takes to run and generate the 18 CSV files.

Measure	Mark
<= 30 secs.	8
>= 30 secs. & <= 40 secs.	6
> 40 secs. & <= 65 secs.	4
> 65 secs. & <= 200 secs	2
> 200 secs	0

Benchmark Test 1b: Accuracy (Size)

The size of the expected output files.

Measured by the number of lines (records) in each station's CSV file (including header line).

Station	No of lines in CSV file
188	107641
203	156921
206	101981
209	5773
213	40783
215	148330
228	6343
270	135643
271	37825
375	92950
395	73245
447	60253
452	114294
459	20902
463	87842
481	1
500	24016
501	10130

Marking:

Measure (+-)	Mark
0%	8 marks
<= 1%	6 marks
> 1%	0 marks

The overall structure, format and commenting of the script file. (4 marks)

Task 2: Data Transformation, Normalisation & XML Validation (20 marks)

Task 2a: Generate normalized XML files (12 marks):

A benchmarking script will access the script ***normalize-to-xml.php*** and run the following tests.

Benchmark Test 2a: Efficiency (Time)
How long the script takes to run and generate the 18 XML files.

Measure	Mark
<= 35 secs.	6 marks
> 35 secs. & <= 65 secs.	4 marks
> 65 secs. & <= 100 secs.	2 marks
> 100 secs.	0 marks

Benchmark Test 2b: Accuracy (Size)
The size of the expected output files.
Measured by the number of lines in each station's XML file.

Station	No of lines in XML file
188	103895
203	156923
206	101983
209	5775
213	37333
215	148332
228	6345
270	135645
271	37827
375	92952
395	73247
447	60255
452	113184
459	20904
463	87844
481	3
500	24018
501	10132

Marking:

Measure (+-)	Mark
0%	6 marks
<= 1%	4 marks
> 1%	0 marks

Task 2b: Design & Implement a XML Schema (8 marks):

- A XSD 1.1 schema developed using the oXygen editor or other (on-line) schema authoring tools.
- A well structured and appropriately strict schema making use of the available syntax and context.
- Your generated XML files (bar the empty one data_481.xml) have to validate against this schema.

Task 3: Charts Visualisation (20 marks)

- Appropriate use of Google Chart (or other) API and toolkit.
- Faithful and accurate rendition of data.
- All code and data files are linked to for examination and marking.
- Code is suitably commented.
- Code is properly attributed where someone else's code is used, with a link to the source.
- Code adheres to good practice in terms of separation of concerns, use of functions etc.

Task 4: Maps Visualisation (20 marks)

- Creative and appropriate use of Google Maps or another mapping API.
- Marks will be based on aesthetics, usability and the creative and accurate use of the data.

Task 5: Reflection & Report (20 marks)

- A short discussion of extending the visualisation.
- A short discussion of XML processing models - DOM oriented parsers (e.g. SimpleXML() as compared to stream oriented parsers (e.g. XMLReader()).
- An outline and discussion of the learning outcomes achieved.

Assessment Offences

This assignment should be your own work. Allowing others to do the work for you, or sharing significant portions of code with others will be considered an assessment offence and may lead to your mark being reduced to 0. Part of the marking process will include similarity checks and we may ask you to explain your code in detail to verify that it is your own. Please refer to the assessment offences policy document for more information.

References

Air Pollution - Wikipedia
UK Government Air Quality Strategy
Markdown Tutorial

---

url: http://fetstudy.uwe.ac.uk/~p-chatterjee/2020-21/modules/atwd2/assignment/