University of Derby **Assignment Specification**

Department of Electronics,
Computing and Mathematics

Module Code and Title: 5CC503 Application Development	
Assignment No. and Title: Assessed Component 2: Data Visualisation Application	
Assessment Tutor: Wayne Rippin	**Weighting Towards Module Grade:** 60%
Date Set: March 29th, 2019	**Hand-In Deadline Date:** Thursday May 9th, 2019 at 4pm

Conditions for Late Submission

Recognising that deadlines are an integral part of professional workplace practice, the University expects students to meet all agreed deadlines for submission of assessments. However, the University acknowledges that there may be circumstances that prevent students from meeting deadlines. There are now 3 distinct processes in place to deal with differing student circumstances:

1. Assessed Extended Deadline (AED): Students with disabilities or long-term health issues are entitled to a Support Plan.
2. Exceptional Extenuating Circumstances (EEC): The EEC policy applies to situations where serious, unforeseen circumstances prevent the student from completing the assignment on time or to the normal standard. http://www.derby.ac.uk/eec
3. Late Submission: Requests for late submission will be made to the relevant Subject Manager in the School (or Head of Joint Honours for joint honours students) who can authorise an extension of up to a maximum of one week.

Work that is submitted late without being covered under one or more of the above processes to deal with student circumstances will be awarded a failing grade of 0%. You must submit your work; otherwise, a grade of NS (non-submission) will be recorded. Under the University regulations, a non-submission may be considered evidence of non-engagement, which may lead to your being removed from the course.

Level of Collaboration

This is an individual assignment. No collaboration with other students or anyone else is allowed.

Learning Outcomes covered in this Assignment:

2. *Design, implement and test database-driven applications using industry-standard techniques, technologies and best practices*.

The Task

Governments around the world are releasing a huge amount of open data – on everything from spending through to crime and health. It is now possible to compare national, regional and city-wide data from hundreds of locations around the world.

However, the data is often just released in tabular form and this is not always the best way of presenting the data. Your task in this assignment is to pick one dataset from any government open data website and create an application that dynamically provides a visualisation for that data that makes it much easier to understand the impact of the data.

Your application must contain the following:

1. A client component that is either:

    a. A web site. You may make this using just client-side technologies (HTML, CSS, JavaScript, etc) or you may also contain some server-side components written in PHP.

    or:

    b. A Windows application written using any language you wish, but it must provide a graphical user interface. You may use Windows Forms, Windows Presentation Framework (WPF) or any other graphical toolkit for the graphical user interface.

2. A server component that is a web service that implements a REST or SOAP API that must be used by your client. The web service must retrieve the data to be visualised from a relational database. Data must be transferred between the client and web service in either JSON or XML format.

3. The visualisation must be generated dynamically at run-time using data supplied by the web service.

4. All server components (both the web service and any server components of the web site if you create a client web site) must be written in PHP and must be published on the Ubuntu virtual machine running on the Azure resource group that has assigned to you for this module.

5. If you provide a Windows client, you must provide a URL on your virtual machine from which a user can download the client executable and any other required files packaged inside a zip file. You may assume that it will run on a machine that has Visual Studio 2017

installed (so you do not need to provide a copy of the .NET Framework or a C runtime system), but any other component required must be included in the zip file.

You are free to provide any form of visualisation of the data source you choose to use. To give you some ideas, here are some possibilities, but you are not restricted to these:

- A game that varies its behaviour depending on the data in the data source;
- If the data is geographical data, you might choose to display it as locations on a map;
- You could choose to represent it graphically on a chart or other form of graph.

You must also include with your submission a report about your site that includes the following:

- The URL of your web site or where the client can be downloaded from;
- The URL of the source of the original data;
- A description of any particularly innovative features of your application or the visualisation created;
- An implementation log or diary that details the work done on this assignment. Each entry in the diary should be in the following format:

Date and time work started on this session.	The amount of time spent working on the application in this session.	A description of the work done in this session. If problems are encountered, you should include a brief description of the problem and how you overcame it. You should also describe how you have tested your work.

Suitable Data Sources

Some of the possible data sources for this assignment can be found at http://datacatalogs.org/. For UK specific data, you can look at http://data.gov.uk. You are also welcome to use other open datasets if you wish. If you have any doubt about the suitability of a dataset, check with the module leader.

You will need to create the relational database and import the data to be visualised into it. You may use third-party tools to do this if you wish. For example, if the data is supplied in XML format or Excel spreadsheets, you might find downloading a trial version of a tool such as Navicat useful (http://www.navicat.com). Note that you are free to use any database engine you wish, but if you use one other than MySQL, you will need to install it on your virtual machine.

Third-Party Libraries

Use of third-party libraries (such as graphics libraries) is acceptable, but their use must be acknowledged in your report.

Submission

By the due date, a zip file containing your assignment MUST be submitted electronically via Course Resources. Full information on doing this can be found at http://www.derby.ac.uk/esub. The assignment must also be published on your virtual machine running on Azure by the due date. You are welcome to publish your application elsewhere if you wish, but it MUST be published to your Azure virtual machine and the URL for the site included in your report. Please name your submission as <your student number>.zip (e.g. if your number is 012345678, your submission would be 012345678.zip).

The zip file must contain the following:

- A folder containing all of the source code for your application, including both the server and client code;
- Your report in either .docx (Microsoft Word) or .pdf format.

Failure to follow these submission requirements will result in your grade being reduced by 20%.

Assessment Criteria

Your submission will be assessed using the following criteria:

Percentage Mark	**Mark Descriptor**
80 – 100%	Outstanding An innovative approach to the visualisation of the data has been used. The code has been written to an excellent standard with no obvious errors and demonstrates significant independent study. An excellent level of testing of all functionality has been performed, as evidenced by the report. As well as providing a visualisation of the data, the data can easily be amended via the application and the changes immediately reflected in the visualisation. An outstanding quality of presentation.

70 – 80%	Excellent An excellent approach to the visualisation of the data has been used that demonstrates some originality. The code has been written to an excellent standard with no obvious errors. A high level of testing of all functionality has been performed, as evidenced by the report. As well as providing a visualisation of the data, the data can easily be amended via the application and the changes immediately reflected in the visualisation. A very high quality of presentation.
60 – 69%	Very good A somewhat obvious, but extremely well implemented visualisation mechanism has been used. The code has been written to a high standard, but there may be one or two obvious issues. The testing is extensive, as evidenced by the report, but there may be one or two omissions. A very good quality of presentation. As well as providing a visualisation of the data; the data can easily be amended via the application, but the changes may not be immediately reflected in the visualisation.
50 – 59%	Good A good choice of visualisation of the data. The code is generally well written, but there are some obvious issues. Most functionality has been tested, as evidenced by the report, but there are quite a few omissions. A good quality of presentation. No mechanism for editing the data has been provided or it is superficial.
40 – 49%	Satisfactory A satisfactory visualisation mechanism has been chosen. Overall, the code works, but there are quite a few problems with it and there is very little evidence of any significant testing. A basic level of presentation. No mechanism provided for editing the data. The web service does not correctly implement a SOAP or REST API.
35 – 39%	Unsatisfactory A serious attempt has been made, but there are significant issues at runtime. The report is superficial and does not provide any evidence of testing. A poor standard of presentation.

1 – 34%	Poor Below the pass standard. Very little has been produced and it is not close to being functional. The report is superficial.