ISIT207/MTS9207 Major Project

The major project for this subject is to create a faceted search application. you have already been working on parts of this project in most of the laboratory exercises. A faceted search application requires a collection of items. The purpose of the application is to support searching via filtering of the collection.

There are two project levels: standard and advanced. An advanced level submission can receive a maximum mark of 16, while a standard level submission can receive a maximum mark of 12.8 (80% of 16). The project is marked in two parts: a demonstration and possible audit in the lab and a via a written report. The hardcopy report is submitted to your tutor when you demonstrate your application in your lab. A softcopy is also uploaded to Moodle. Advanced projects are marked in the week 11 lab. Standard projects are marked in the week 12 lab. The requirements for an advanced project include all the requirements for a standard project.

Requirements for a Standard Project

1. Supports three categorical attributes. Note a numeric attribute can be converted into a categorical one by replacing a number with an interval. If unsure about this, ask for guidance.
2. Supports one numeric attribute. This attribute will be filtered using two input fields (or similar) that allow a user to enter a numeric range as a lower bound and upper bound.
3. The initial state for the web page is: all filters are unset and the entire table of items (i.e. the whole collection) is shown.
4. The web page shows a filter for each categorical attribute. A filter could be implemented as a list of radio buttons or a list of checkboxes or some equivalent, where each element in the list is for an attribute value. For example, for an attribute colour filter the list could be for 'red', 'green' etc. Note the user must be able to see all values at a glance without any keyboard or mouse interaction, so a dropdown menu will not be acceptable.
5. Each filter also shows counts for each attribute value. For example, 'red (5)', 'green (8)' which indicates there are 5 red items and 8 green items in the filtered table.
6. The web page supports filtering by multiple attributes. A user can make a selection in one or more filters then click a filter button. After the click the item table will be redisplayed to only show items that meet the filter conditions. The counts next to each attribute value in each filter are updated after the click.
7. There will be a Reset button that clears all filters.
8. The item collection will be loaded from a CSV file that is located on a server. The item collection will contain at least 50 records. A local server is fine.
9. Each time a user clicks the filter button, filter settings are saved. Later a user can redo a prior saved filter by selecting from a list of prior filter actions. If the user exits the browser then later revisits the web page, prior filter actions will still be available.
10. Reasonable use of styles i.e. CSS must be made.
11. Numic inputs must be validated and a reasonable response to web page users made if input is not valid.
12. Your report will (i) describe what your faceted collection is and (ii) provide sufficient use cases communicated via screen shots and text descriptions that prove your application meets all the above (and below if advanced) requirements. See report requirements on the next page for further details.
13. Your code will be maintainable. A minor change to the requirements should require a minor coding effort to implement. If unsure about this, ask for guidance.

Additional Requirements for an Advanced Project

1. Supports an additional text attribute that contains a description i.e. free text. Web page contains an additional filter that supports keyword search over this extra attribute.
2. Each categorical filter supports selection of multiple attribute values, for example, colour = {red or blue}.
3. The web application is parameterised. This means there will be no reference to your attribute names or values in your HTML or JavaScript the only reference to them will be in your item collection file or files. This will be tested by adding/removing columns and by adding/removing values from your item collection then checking if your application still works.

Marking scheme: standard

The marking scheme for standard has two parts: features and the report.

Features (*8 marks*)

1. Loads the collection file from a server (local) and shows the full table on start-up. *2 marks*.
2. Filtering works as per requirements. *4 marks*.
3. The search history works as per specification. *2 marks*.

Report (*4.8 marks*)

1. Appropriate use of CSS to make the page look reasonable. *1.8 marks*.
2. The clarity of the test cases. *1.5 marks*.
3. The maintainability of the code. *1.5 marks*.

The total mark is: feature_mark + (report_mark * feature_mark / 8) i.e. the report mark is weighted by the amount of features that are implemented.

Marking scheme: advanced

Features (*3.2 additional marks*)

1. Advanced requirement 1. *0.6 marks*.
2. Advanced requirement 2. *0.6 marks*.
3. Advanced requirement 3. *2.0 marks*.

Report requirements

The report will provide sufficient detail to demonstrate that your programs works i.e. meets requirements. If the tutor does not have time to fully text (i.e. check) your program he will rely on the test/use cases included in your report. A test/use case can start with a description of the requirements being met, a description of the user interaction plus screenshots that show how your program behaves. Your report should include a page that shows your faceted search items in a simple table to help with this. Your report will include the source code for filtering i.e. the code that is executed when a user clicks the filter button to update/create the filtered table of items. You will also upload a zip file that contains all your software and supporting files plus your report to a Moodle drop box. If a feature is missing from your report, your tutor will assume it has not been implemented, so take care to include what you have implemented in your report. For advanced projects, the hardcopy report should be handed to your tutor before you are marked (so your tutor can annotate it), and the softcopy uploaded to Moodle just after you have been marked (unless your tutor gives other directions). For standard projects the hardcopy should be handed to your tutor at the start of your lab, and softcopy should be uploaded to Moodle at the start of the lab. We expect all advanced level software to be checked in the lab in week 11, while most/some standard level software will be checked in the lab (hence the different requirement regarding the hardcopy and softcopy submissions) in week 12.