

Criterion	7	6	5	4	3	2-1
Application & Usability 10 + 15 + 5 = 30 marks	<p>The application presents a compelling integration of at least two distinct data sources and/or services, where the integration allows a far richer service experience than could be achieved by separate use of the APIs.</p> <p>The application is robust and executes without noticeable error and handles gracefully any service failures. Deployment through the Docker container is seamless and error free, and there are no configuration issues for the required environment. .</p> <p>The application is clean and uncluttered, and the principal use cases are a natural fit for the user interface, allowing execution without undue effort or confusion.</p>	<p>The application presents a compelling integration of at least two distinct data sources and/or services, where the integration allows a somewhat richer service experience than could be achieved by separate use of the APIs.</p> <p>The application is robust and executes without noticeable error and handles gracefully most service failures. Deployment through the Docker container is well organised and largely straightforward, although some minor configuration adjustments may be required.</p> <p>The application is clean and generally uncluttered, and the principal use cases are readily executed, though there may be some clumsiness in the workflow.</p>	<p>The application presents a coherent integration of at least two distinct data sources and/or services. The integration adds some value but the service enhancements are restricted in scope or limited by mismatches between the services.</p> <p>The application is robust and executes with only occasional errors and handles gracefully most service failures. Deployment through the Docker container is basically successful, but may require significant effort</p> <p>The application is clean and generally uncluttered, and the principal use cases are reasonably easily executed, though there are some questionable choices in the design.</p>	<p>The application presents a plausible integration of at least two distinct data sources and/or services, but the benefits are marginal, or the expanded queries are poorly handled by the other services.</p> <p>The application basically works, but the robustness is questionable and there are obvious errors and service failures are not handled well. Docker deployment has major issues.</p> <p>The application is somewhat clean and uncluttered, but overall the design is clumsy and doesn't operate very well.</p>	<p>The application integrates the required number of sources and services but the advantages are minimal or service mismatches severely limit its value.</p> <p>The application basically works, but there are numerous errors and failures. Docker deployment is markedly incomplete.</p> <p>The application is poorly laid out, but the basic functionality remains usable.</p>	<p>The application does not meet the specification or is fundamentally flawed in its use of the data sources and services.</p> <p>The application is markedly incomplete and results in numerous errors. Docker container deployment is not attempted.</p> <p>Layout is poor and usability marginal.</p>
Marks (30)						

Development Process & Code Quality 10 marks	The application shows clear evidence of a professional approach to development, with a coherent application architecture, documented usage of the APIs and code quality at a professional level.	The application shows good evidence of a professional approach to development, with a coherent application architecture, documented usage of the APIs and code quality at a professional level other than some minor variations.	The application shows some evidence of a professional approach to development, with a reasonably complete application architecture, documented usage of the APIs and competent, but not always professional level code quality.	The application shows limited evidence of a professional approach to development, with an ad hoc or unclear application architecture, similarly ad hoc and suboptimal usage of the APIs and variable level code quality.	Application development has been ad hoc and little more than hacking, with no obvious architecture. API usage is poorly documented and poorly chosen, albeit functional in some respects. Code exhibits numerous defects when compared to the standard expected.	Application development is deeply flawed, with no architecture and poor code quality.
Marks (10)						
Report and User Guide 10 marks	The report is thoroughly professional and addresses each of the listed requirements in detail and with only occasional errors of grammar or structure.	The report is thoroughly professional and addresses each of the listed requirements in detail and with some more frequent errors in grammar or structure.	The report is professional but lacks some detail in a small number of the listed requirements. There may be occasional errors of grammar or structure.	The report is adequate, but the coverage is deficient in a number of the listed requirements. Grammar and structure may be somewhat variable, but are overall ok.	The report is somewhat adequate, but the coverage is deficient in many of the listed requirements. Grammar and structure are of variable quality.	The report is flawed and doesn't meet the requirements. There may be whole sections missing or poorly covered. There is no coherent professional report structure as required
Marks (10)						
<b>TOTAL (50)</b>						