






Mark range %	100-80%	79-60%	59-40%	39-20%	19-0%
Assessment Criteria	<b>"A system ready to deliver to go live"</b> 	<b>"A usable system with useful features"</b> 	<b>"A minimum viable system"</b> 	<b>"Basic data/items listings"</b> 	<b>"Non-functioning system"</b> 
Use of Bootstrap with appropriate navigation features (10 Marks)	Considerable amount of code demonstrating very good use of Bootstrap (or other grid system)	Reasonable amount of code demonstrating good use of Bootstrap (or other grid system)	Reasonable amount of code demonstrating some use of Bootstrap (or other grid system)	Small amount of code using MVC correctly, and basic template layout.	No MVC or template layout
Well designed and appropriately populated database tables (10 Marks)	Appropriate number of tables, with relevant fields and types, complex cross-referenced using foreign keys, joins, and each populated with a large number of useful realistic records and image references (1000s)	Appropriate number of tables, with relevant fields and types, cross-referenced using foreign keys, joins, and each populated with a large number of records (1000s)	Some tables, with relevant fields and types, cross-referenced using foreign keys, and each populated with a reasonable number records (100s)	A few independent tables, with relevant fields and types, each populated with a few records (10)	Main table, with relevant fields and types, and populated with a few records.
Well designed OO code using MVC Design Pattern, with appropriate names installed on university servers (20 Marks)	Considerable amount of OO code demonstrating correct use of MVC, classes/properties and methods, using PDO to access the database. Excellently commented code. Demonstrable performance enhancements. Installed and running on the university web server and using university database server.	Reasonable amount of OO code demonstrating correct use of MVC, classes/properties and methods, using PDO to access the database. Well commented code. Implementation of performance enhancements. Installed and running on the university web server and using university database server.	Small amount of OO code demonstrating correct general use of MVC, classes/properties and methods, using PDO to access the database. Some code comments. Consideration of performance.	Small amount of OO code demonstrating limited understanding of MVC, classes/properties and methods, or database access. Basic code comments.	Very little code demonstrating no real understanding of MVC, classes/properties and methods, or database access. Little or no code comments.
Ability to create self-registration, records and login (10 Marks)	Users can register themselves and login successfully using encryption for their passwords. Anti-spam feature used. Some form of session used to maintain state. Malicious code filtering on items. Driven by excellent, logical OO design. Excellently commented code.	Users can register themselves and login successfully using encryption for their passwords. Some form of session used to maintain state. Validation on input fields. Driven by good, logical OO design. Well commented code.	Users can register themselves and login successfully. Some form of session used to maintain state.	Registration and login not fully implemented/working correctly	Registration and login not implemented, or not working
Ability to display information from records stored in the database (20 Marks)	Users can retrieve and display user details appropriate to their role, including images. Sophisticated, responsive layout using CSS/Bootstrap. Includes paging and web scraping protection. Demonstration of system with large number of realistic records/users – you can generate this data. Driven by excellent, logical OO design. Excellently commented code.	Users can retrieve and display user details appropriate to their role, including images. Good responsive layout using CSS/Bootstrap. Demonstration of system with large number of records/users – you can generate this data. Driven by excellent, logical OO design. Well commented code.	Users can retrieve some item details. Basic cellular listing layout using tables or divs with some formatting evident. Attempting display of individual items. Some code comments.	Users can retrieve some item details. Simple table or list of text data, Basic code comments.	Users cannot retrieve or display details from the database. Little or no code comments.
Ability to search for users and create/edit friends. (10 Marks)	Comprehensive, interactive faceted search to narrow down output. Complex SQL statements. Driven by excellent, logical OO design. Excellently commented code.	Free text search facility with 2 or 3 extra filters combined into search. Basic OO implementation. Well commented code.	Free text search facility working but basic. Some filters implemented, but may have issue. Some code comments.	Basic text search facility may have issues. Basic code comments.	Some search/filters partly implemented or no implementation. Little or no code comments.
Ability to create/edit friends. (20 Marks)	Users can add, view and edit items in their "friend list" – including the friend confirmation process and notification of friend requests. Driven by excellent, logical OO design. Excellently commented code.	Users can add, view items in their "friend list" – including a basic but functional friend confirmation process. A Basic OO implementation. Well commented code.	Attempt at "friend list" and process. Some code comments.	Basic "fiends list" feature but has issues.. Basic code comments.	Little or no implementation. Little or no code comments.