|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mark range %** | **100-80%** | **79-60%** | **59-40%** | **39-20%** | **19-0%** |
|  | Image result for yellow gallardo white background **“An auction system ready to deliver to go live”** | Related image **“A usable auction system with useful features”** | **Image result for vauxhall white  background  “A minimum viable auction system”** | Image result for nissan micra 1998 **“Basic data/items  listings”** | **“Non- functioning system”** |
| Completion of online tests (10 Marks) | Up to 10 marks available cumulatively across 3 Blackboard based Tests covering the material discussed in Trimester 1. HTML/CSS/PHP questions. Various deadlines spread over Trimester 1. | | | | |
| 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 (20 Marks) | Considerable amount of OO code demonstrating correct use of MVC, classes/properties and methods, using PDO to access the database. Well commented code. Demonstrable performance enhancements. | 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. | 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. | | 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 auction and item details, 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. | Users can retrieve and display auction and item details, including images. Good responsive layout using CSS/Bootstrap. Includes paging. Demonstration of system with large number of records/users – you can generate this data. Driven by excellent, logical OO design. | Users can retrieve some auction/item details. Basic cellular listing layout using tables or divs with some formatting evident. Attempting display of items and replies. | Users can retrieve some item details. Simple table or list of text data, | Users cannot retrieve or display details from the database |
| Ability to search auctions and items and place bids  (20 Marks) | Comprehensive, interactive faceted search to narrow down output. Complex SQL statements. Users can add, view and edit items in their “bidding list” – notification of outbidding. Driven by excellent, logical OO design. | Free text search facility with 2 or 3 extra filters combined into search. Users can add, view items to their “biding list”. Basic OO implementation. | Free text search facility working but basic. Some filters implemented, but may have issue. Attempt at “bidding list” | Basic text search facility may have issues. | Some search/filters partly implemented or no implementation |