CMPS 356 Enterprise Application Development - Spring 2018

**Lab 6 –Asynchronous JavaScript and Unit Testing**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Student Id** |  |
| **Email** |  |

**Grading Rubric - - In the *Functionality* column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | % | Functionality\* | Quality of the implementation | Score |
| Implemented the following methods inside BooksRepo.js |  |  |  |  |
| * getBook(bookName) | 10 |  |  |  |
| * getBooksWIthPageCountMoreThanX(pageCount) | 10 |  |  |  |
| * getAuthorBooks(author) | 15 |  |  |  |
| * getAuthorsBookCount() | 20 |  |  |  |
| * getBooksbyCatagory(category) | 15 |  |  |  |
| Created a BooksRepo.spec.js file that implements the test cases for all the BooksRepo.js methods | 30 |  |  |  |
| Total | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | -100 |  |  |  |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***Not*** ***working*** (lose 40% of assigned grade) and ***Not done*** get 0. The remaining grade is assigned to the quality of the implementation. In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation. Quality includes meaningful naming of identifiers, no redundant code, simple and efficient design, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation. **Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor user interface design.

## Testing evidence (You should provide a screenshot of your work)