CMPS 356 Enterprise Application Development - Spring 2018

**Lab 10– React JS – Web UI**

|  |  |
| --- | --- |
| **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 |
| 🡺Home Component  Implemented all requirements for the home page as mentioned in the lab sheet and shown in the demo video.  The page should allow users to be able to  🡪 display all the available books by category, name, isbn, author or pageCount  Also the page should have the following links  🡪 update link  🡪 delete a book  🡪Details link  🡪Borrow a book | 30 |  |  |  |
| 🡺Borrow a book component  A form that allow users to borrow a book . It should be similar to the demo shown in the video | 15 |  |  |  |
| 🡺 Details component  Show the details of the book when the details link is clicked | 15 |  |  |  |
| 🡺BookForm Componenet  Provides a form to allow adding a book to the book-catalog. The entered book should be posted to the Web API @ /api/books/  The form can also be used for updating an existing book. The updated book should be sent to the Web API @ /api/books/isbn | 20 |  |  |  |
| 🡺BooksSummary Component  Display all the different book summaries such as   * Displaying the Books * Top 3 Borrowers * Top 3 Borrowed Books * Summary for Borrowed Categories | 20 |  |  |  |
| 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