CMPS 356 Project Phase 2 – Web API and Data Management using MongoDB (10% of the course grade)

The project phase 2 submission is due by 8am Sunday 8th May 2022

1. Deliverables

You are to complete the implementation of phase 1 Web App by delivering the following:

1. Address the project phase 1 feedback given during the demo session and the requested corrections in the grading sheet. You may also use phase 1 model solution provided as a base for phase 2.
2. Design and implementation of the database schema to manage the data in a MongoDB database (include a schema diagram in the project report). The implementation should use mongoose.
3. Populate the database with the data from the json files.
4. Implement the repository methods to read/write data from MongoDB as the data source. The *repositories* should offer the same functionality as phase 1. All **data filtering should be done on the server** using MongoDB queries and only the required data should be retrieved. The implementation should make use of MongoDB query capabilities (e.g., using aggregate query to get the data for the reports).
5. Implement Web API to make to app functionality remotely accessible via HTTP.
6. Update the Web UI implementation to use the Web API (instead of IndexedDB).
7. Design and Testing Documentation

- Document 4 **technical** lessons learned by comparing your submitted project phase 1 with the model solution provided. You need to provide detailed reflections about the new concepts and technical lessons learnt.
- Document MVC architecture diagram for your overall design.
- MongoDB database schema diagram.
- Write a testing document including screenshots of conducted tests illustrating a working implementation.
- Every team member should submit a description of their project contribution. Every team member should participate in the solution demo and answer questions during the demo.

Important notes:

- Continue posting your questions to https://github.com/cmps350s22/cmps350-content/issues.

- Do not forget to submit your design and testing documentation (in Word format) and fill-up the Functionality column of the grading sheet using the provided phase 2 template.

- Push your implementation and documentation to your group GitHub repository as you make progress.

- You need to test as you go!

- Seek further clarification about the requirements/deliverables during office hours or by posting questions online. Note that further important clarifications maybe added to this document, and you will be notified.
- Report any team issues early, any issues reported towards the submission deadline will be ignored.

2. Grading rubric

Criteria	%	Funct Ionality*	Quality of the implementation	Grade
Addressing the project phase 1 feedback	5			
Design and implementation of the database schema to manage the data in a MongoDB database (including a schema diagram in the project report).	10			
Populate the database with the data from the json files.	5			
Repository Implementation to read/write data from/to MongoDB	30			
Web API implementation	25			
Change Web UI to use the Web API.	20			
Design and Testing Documentation *** Design documentation:** - 4 key lessons learned from Phase 1. - Web UI services diagram - MVC architecture diagram. - Database schema diagram. *** Testing documentation:** with evidence of working implementation using snapshots illustrating the results of your solution testing (you must use the provided template). * **Discussion of the project contribution** of each team member [-10pts if not done]	10			
Total	100			
Copying and/or plagiarism or not being able to explain or answer questions about the implementation.	0			

* **Possible grading for functionality**: ***Working*** (get 60% of the assigned grade), ***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. Solution quality also 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 for code duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission and unnecessary complex/poor user interface design.