**CS649  
Assignment 4**

In this assignment, you will build on your previous assignment by taking advantage of MongoDB and other tools. Your job will be to eliminate the array-based storage of your application and instead, create a new MongoDB database and collection. Then, you will reconfigure your inventory management application to begin creating data within and retrieving data from your MongoDB database.

**General Points to Consider**

* As was outlined in the book, restructure your application to take advantage of an api folder and a separate ui folder. Make sure to configure the appropriate package.json files and install the corresponding dependencies within each directory. The server should run at localhost:3000 while the ui should be running at localhost:8000.
* Reconfigure your project to use global variables via .env. URLs and port numbers should all be reconfigured now to use global variables DB\_URL and API\_SERVER\_PORT respectively. Make sure that you have a .env file to read these values from in both api and ui folders. Again, you will need to install the appropriate dependencies and configure each file in your application accordingly.
* Reconfigure your application to store data within MongoDB. When the user adds a new product, the category, product name, price, and image path (string) should be created within MongoDB. Remember, MongoDB will automatically create a unique identifier for you. Even so, you still need to store the ID that is part of your schema and figure out a way to increment that within the collection.
* Reconfigure your application to retrieve data from MongoDB. Once you’re confident that your connections to your MongoDB instance are successful, remove the array code from your application entirely.
* Install and configure ESLint. Lint your code and fix all errors before submitting your assignment.

*Note: Although we covered WebPack in this module, you will not be required to configure your project to use it in this assignment.*