

**COM S/SE 319: Construction of User Interfaces**  
**Spring 2023**

**Assignment 03: MERN**

[Total Points: 100]

**Assignment Due: Sunday, April 30th, 2023, 11:59PM**

**1. Overview**

In this assignment you will develop a MERN application. MERN is a Web application developed using MongoDB, Express, React and Nodejs.

- Mongo
- Express
- React
- Nodejs

In your application you will use the catalog of products from the “<https://fakestoreapi.com/products>” site. You have utilized this catalog before, and you should be familiar with this information.

The application must execute the next Request Methods:

- Post to Create - adding one product at a time.
- Get to Read - this functionality will show all available products.
- Put or Post to Uupdate – it will allow us to modify at least one property of the product, e.g., price.
- Delete to Ddelete - for deleting one product given its id property.

For the sake of simplicity, you will select a minimum of **three products** from the catalog to develop all the Assignment02. And there is no penalty for using more than three products.

The catalog of the products **MUST** be in MongoDB. It is strictly prohibited using a JSON file or other to substitute the Mongo database.

Your client will have four views in one Single web page development :

1. A view to add or create a new product to the database in Mongo.
  - a. A form must be implemented to add the information.
  - b. The data to include per product is:
    - i. \_id
    - ii. Title
    - iii. Price
    - iv. Description
    - v. Category
    - vi. Image (read below about it)
    - vii. Rating

1. Rate
2. Count
- c. This addition is obtained using Post method.
- d. You don't need to invent the information; you just need to read the data from the ["https://fakestoreapi.com/products"](https://fakestoreapi.com/products) site.
2. A view to show or read all the products contained in your catalog in Mongo.
  - a. This view is obtained by using the Get method.
  - b. You should arrange the products in an aesthetic view using Bootstrap or Tailwind.
  - c. It is optional, but highly recommended, to Get the information of one particular product given its **id**.
3. A view to uppdate the **p**rice for any product already contained in the catalog in Mongo.
  - a. First, you must enter the product **id** and then show the full information of that product.
  - b. Second, using an input field you can enter a new **p**rice data to be updated in the Mongo database.
  - c. The update functionality must be done just to one product
4. A view to delete one product from the catalog in Mongo.
  - a. . First, you must enter the product **id** and then show the full information of that product.
  - b. Second, using a button you can Click to remove the product from the catalog in Mongo.
  - c. It is optional, but highly recommended, to have a second question to confirm the certainty of deleting the product from the catalog.

All functionalities must be developed by a client Frontend developed in React. Otherwise, the Assignment will be highly penalized.

The backend, the web server must be developed in Nodejs using Express. Otherwise, the Assignment will be highly penalized.

The storage of the catalog must be implemented in MongoDB. Otherwise, the Assignment will be highly penalized.

Implementing Mongo database can be done by two options:

- a. Using MongoDB driver
- b. Using Mongoose

Any of these options are permitted.

This time Mongo database must be implemented in the local disk installing the version Mongo Community or Mongo Compass

This is an assignment that can be developed in Team. Otherwise, if you prefer to develop this assignment alone, please notify the Professor to review it and grade it individually.

Feel free to add small touches to the webpage to enhance the view. You can use:

- Bootstrap icons, components etc.
- Feel free to play with font sizes, colors, numeric values.
- You can customize the browse view, table and form components however you like. But make sure you display all stated fields and values.

- Custom CSS to:
  - Enhance the look and feel with custom coloring, components, fonts etc.
- Just remember to maintain the underlying bootstrap/tailwindcss theme.

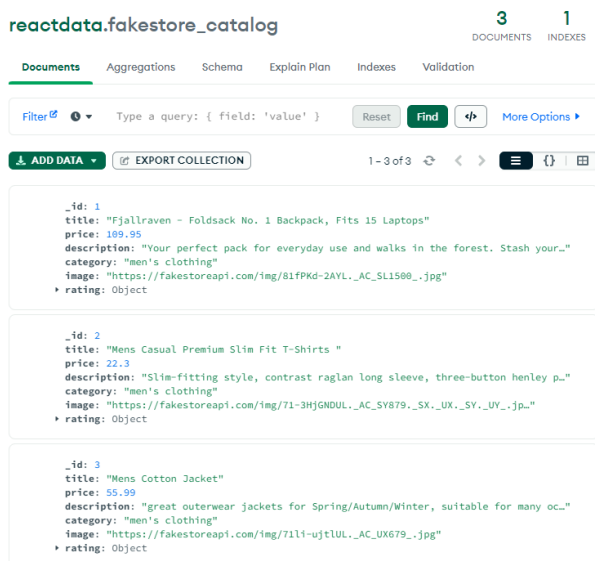
## 2. Images.

The images must be in a folder `./images` in the Backend. Therefore, the URL of every image will be relative to the backend <http://host:port> location. To have the image in the image folder you must copy it using copy paste operations; that is, to add image files to our system, this time, won't be using the MERN application but using manual operation.

## 3. MongoDB initial load.

The initial load of the data into Mongo database will be done by copying the selected products and pasting to Mongo Compass interface.

- You will create a database with the next name:  
**reactdata**
- You will create a collection with the next name:  
**fakestore\_catalog**



## 4. The user interface – menu type.

The option to select the Method Create, Read, Update or Delete products must be implemented in a menu to allow the user to select the desired option. The menu can be a navbar, simple buttons, input field, or any other technique.

## 5. Credits.

Add a view to show the students information. The view should contain complete correct names, email, of every team member. Additionally, you should add the next information: course number, course name, date, professor name and a brief paragraph of two-three lines introducing your project.

## 6. What to Submit:

This assignment implies creating a React application and a Nodejs application. It is recommended to have React and Nodejs in different folders with the same parent folder. When you install React or Nodejs, a node\_module folder is created with a lot of libraries inside causing many megabytes of code. **It is not necessary to add these node\_modules folders to the file which will be submitted to Canvas.**

This time you will create TWO Zip files and a short video :

- One Zip file which contains all the files developed to the front-end – `TeamXXA3frontend.zip`
- One Zip file which contains all the files developed to the back-end – `TeamXXA3backend.zip`
- **Video** of 1.5 to 3 minutes showing your frontend files, showing your backend files and the application in execution. Include briefly and quickly the execution of the four operations: Create, Read, Update and Delete and the Mongo view. The importance of the video is allowing the reviewer/grading to observe clearly the evidence of the MERN application. On one side, a quick view of the front-end files and back-end files show the structure of React and NodeJs. On the other side, the application running is evidence of the correct functioning. You must show the content of the Mongo database before and after every Request Methods.  
Name this video – `TeamXXA3video.mp4`

The files that you must add in the Zip file containing the front-end include :

- `index.js` <<- the name can be `app.js` also or any other.
- `package.json`
- [all additional Javascript files or CSS style files used to develop components]
- Any other file such as images added to frontend (not including catalog of products)

The files that you must add in the Zip file containing the back-end include :

- `index.js` <<- the name can be `app.js` also or any other.
- `dataSchema.js` <<- Mongoose dataschema
- `./images` <<- images pertaining to the product catalog.
- `package.json` <<- containing init and dependencies installed.
- [optionally your `style.css` if you have one]

**7. The Rubric to grade this assignment is as follows:**

**This Assignment has a total value of 100 points.**

**Late submission will deduct 5 points per day late.**

**Front-end 30 points :**

- Has a Create view which includes form to input data ? \_\_\_\_\_ / 10
- Has a Read view which shows all / id products ? \_\_\_\_\_ / 5
- Has an Update view which includes one product to modify ? \_\_\_\_\_ / 5
- Has a Delete view which includes one product to delete ? \_\_\_\_\_ / 5
- Has a friendly easy to use interface Single web page to select C/R/U/D ? \_\_\_\_\_ / 5

**Back-end 60 points:**

- Has an `app.get (` that connects to mogo database to read products ? \_\_\_\_\_ / 15
- Has an `app.post (` that connects to mongo database to add products ? \_\_\_\_\_ / 15
- Has an `app.post (` or `app.put (` that connect to mongo db to update products ? \_\_\_\_\_ / 15
- Has an `app.delete(` that connect to mongo db to delete a product ? \_\_\_\_\_ / 15

**Video 10 points:**

- It includes the video as required. \_\_\_\_\_ / 10

**### TOTAL \_\_\_\_\_ / 100**