

**COM S/SE 319: Construction of User Interfaces**  
**Spring 2024**

**Assignment 03: MERN Application Development**

[Total Points: 100]

**Assignment publishing:** Monday April 15<sup>th</sup>, 2024

**Assignment Due:** Saturday, April 27<sup>th</sup>, 2024, 11:59PM

(two weeks from the date of publication)

**Objective:**

Develop a MERN (MongoDB, Express, React, Nodejs) application for managing a catalog of items using the "https://fakestoreapi.com/products" dataset. Implement key CRUD functionalities and ensure a well-organized, user-friendly interface.

**Requirements:**

**1. MERN Application Development**

In this assignment you will develop a MERN application. A MERN stack uses 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.

**Request Methods:**

The application must implement the next Request Methods:

- Post : **C**reate - Add one product at a time.
- Get : **R**ead – Display all available products and one item selected by it id.
- Put : **U**ppdate – Modify at least one property of the product (e.g., price)
- Delete : **D**eleate – Remove a product given its id property.

**Product Selection:**

For the sake of simplicity, you will select a minimum of **six products** from the catalog to develop the application. There is no penalty for using more than six products.

## Data Storage:

Strictly use MongoDB for the product catalog; it is prohibited to use JSON files or other substitutes.

## Frontend Development:

Your client will have four views in one Single web page development. And one view for the student and course information. In total five views are implemented in one single web page.

1. A view to add or create a new product at the time to the database in Mongo.

Implement a form to add a new product to the MongoDB.

The form must use Bootstrap or Tailwind CSS style.

The data to include per product is:

- id
- Title
- Price
- Description
- Category
- Image (see below)
- Rating {Rate, Count}

Use the **Post method** to add products.

You don't need to invent the information; read the data from the "<https://fakestoreapi.com/products>" site.

2. A view to show or read all 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. This same view must implement getting the information of one particular product given its **id**. This is the case to consult one item by its id using GET access method.
3. A view to uppdate the **price** 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 before to proceed the modification. Once the product information is confirmed, the modification is executed.
  - b. Using an input field, you can enter new **price** data to be updated in the Mongo database and a button to proceed.
  - c. The update functionality must be done just to one product at a time.
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. Once the product information is confirmed, the deletion is executed.
  - b. Using a button, you can Click to remove the product from the catalog in Mongo.

- c. Must be a second input question to proceed the deletion.

## 5. Student Information View:

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.

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

## Backend Development

Develop the web server in Nodejs using Express; failure to do so will result in significant penalties.

Implement MongoDB storage using either the MongoDB driver or Mongoose.

Implementing Mongo database can be done by two options:

- Using MongoDB driver
- Using Mongoose

Any of these options are permitted.

This time Mongo database must be implemented in the local disk installing the version Mongo Community. Install Mongo Compass to manage the product catalog using mongo shell if necessary and to review the result of the access methods.

Implement the methods :

- app.get
- app.post
- app.put
- app.delete

## Teamwork:

This is an assignment that can be developed in Team. If choosing to work individually, notify the professor for individual review and grading.

## Enhancements:

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

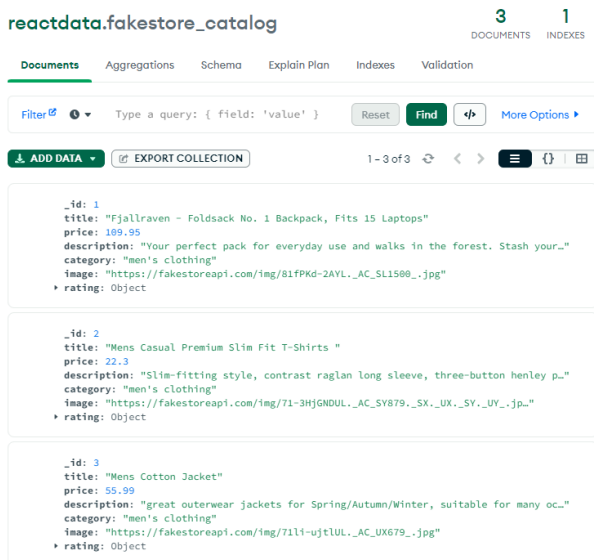
- Bootstrap icons, components etc.
- Feel free to play with font sizes, colors, numeric values.
- You can customize the browse view, table and form components however you like. But make sure you display all stated fields and values.
- Custom CSS to:
  - Enhance the look and feel with custom coloring, components, fonts etc.
- Just remember to maintain the underlying bootstrap/tailwindcss theme.

## 2. MongoDB initial load.

### Data loading:

The initial load of the data into Mongo database will be done by a manual operation 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**



### 3. The user interface – menu type.

#### Method Selection:

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.

### 4. Video file :

Produce a video with a duration ranging from 1.5 to 3 minutes, showcasing the functionality of your MERN application. The video should provide a concise overview of your frontend and backend files, demonstrating the execution of key operations such as Create, Read, Update, and Delete.

Additionally, emphasize real-time modifications in the MongoDB database corresponding to each Request Method.

The video serves a crucial role in enabling reviewers to clearly assess the MERN application's performance. It offers a rapid glimpse into the structural aspects of React and NodeJs through a brief presentation of front-end and back-end files.

Simultaneously, the running application serves as tangible evidence of its correct functionality. Ensure to display the Mongo database content both before and after the execution of each Request Method for comprehensive evaluation.

### 5. Submission Requirements:

#### Zip File containing software Frontend and Backend:

This assignment implies creating a React application and a Nodejs application. You will create one folder for the frontend and one folder for the backend:

- Frontend
- Backend

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 Zip file to be submitted to Canvas.**

#### Frontend files:

You will add all the files in the Frontend folder except node\_modules. Examples of these files are:

- index.js
- 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)

#### Backend files:

The files that you must include are :

- index.js <<- the name can be app.js or any other file name
- dataSchema.js <<- Mongoose dataschema if used
- Folder "./images" if used
- package.json <<- containing dependencies installed.
- [all additional style.css if you have any]

#### File name:

Zip file name – `TeamXXAssignment3.zip`

Name this video – `TeamXXA3video.mp4`

Do not include the video in the Zip file.

### 6. 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 ? \_\_\_\_\_ / 5
- 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 and a good aesthetic presentation? \_\_\_\_\_ / 10

#### 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.put()` that connects to mongo db to update products ? \_\_\_\_\_ / 15
- Has an `app.delete()` that connects to mongo db to delete a product ? \_\_\_\_\_ / 15

#### Video 10 points:

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

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

Dr. Abraham Aldaco