



DOUGLAS COLLEGE

COMMERCE AND BUSINESS ADMINISTRATION

CSIS 3380-002: ADVANCED WEB PROGRAMMING WITH JAVASCRIPT & AJAX

## Final Exam (December 16, 2021)

Total Duration: 2 hours 50 minutes

Total marks: 100

### Hands-on Problems (30 + 70 = 100 marks).

#### Instructions

- Begin by downloading the **FinalExam** folder from Blackboard or you can create your own **lastName\_studentID** folder. Then, create a react app named **client** within this folder. Inside the **src** subfolder, the **App.js** file should contain the answer to the **first question**.
- Copy the **styles.css** from the exam folder and place it inside **src** subfolder
- Next, create a **client.js** file within the **src** subfolder. The **client.js** should contain the code (front-end UI) for the **second question**. If you need to create additional **.jsx** files, make a **components** subfolder inside **src** and place all additional components within this.
- Next, create a node server inside the **FinalExam** folder by running **npm init** command. The starting point should be **server.js** file. This file should contain the backend and the database connections for the second question.

#### Submission:

Once you are done, please create two copies: **lastName\_studentID\_withNodeModules** and **lastName\_studentID\_withoutNodeModules** and upload these two folders to Blackboard under **FinalExam**.

**Important:** The **src** folder and **server.js** should contain all of your work for both the questions.

#### Question 1: Retrieving data from an API and displaying it using React (**Place the code in App.js**)

- (a) TheCocktailDB offers a free JSON API that provides lookups for popular and recent Cocktails. Retrieve the details about Margarita drink from the API:

<https://www.thecocktaildb.com/api/json/v1/1/search.php?s=margarita>. Fetch and display the information for different margarita drink types. Process the response to extract the "strDrink", "strCategory", "strGlass", "strDrinkThumb", "strInstructions" and "dateModified". (Use Axios or Fetch API or jQuery AJAX).

You will have to render the image for "strDrinkThumb". Use height = {150} and width = {150} values with **<img>** tag.

**(15 Marks)**

- (b) Display information as shown in excerpt.

- Use the **styles.css** for the displaying the information as shown.
- Refer to the DOM snapshot below for placement of information

**(15 Marks)**

```

▼<div class="App">
  <h1> It's Festive season. Lets have some fun!</h1>
  <h1> Types of Margarita Drinks</h1>
  ▼<div class="drinks"> (flex)
    ▼<div class="drink">
      ▼<h3>
        "Drink "
        "1"
      </h3>
      <h2>Margarita</h2>
      ▼<div class="details">
        ▼<p>
          "Drink Category: "
          "Ordinary Drink"
        </p>
        ▼<p>
          "Glass: "
          "Cocktail glass"
        </p>
        
        ▼<p>
          "Instructions: "
          "Rub the rim of the glass with the lime slice to make the salt stick to it. Take
          care to moisten only the outer rim and sprinkle the salt on it. The salt should
          present to the lips of the imbiber and never mix into the cocktail. Shake the other
          ingredients with ice, then carefully pour into the glass."
        </p>
        ▼<p>
          "Date Modified: "
          "2015-08-18 14:42:59"
        </p>
      </div>
    </div>
    ►<div class="drink">...</div>
    ►<div class="drink">...</div>
    ►<div class="drink">...</div>
  </div>

```

## It's Festive season. Lets have some fun!!

### Types of Margarita Drinks

#### Drink 1 Margarita

**Drink Category:** Ordinary Drink

**Glass:** Cocktail glass



**Instructions:** Rub the rim of the glass with the lime slice to make the salt stick to it. Take care to moisten only the outer rim and sprinkle the salt on it. The salt should present to the lips of the imbiber and never mix into the cocktail. Shake the other ingredients with ice, then carefully pour into the glass.

**Date Modified:** 2015-08-18 14:42:59

#### Drink 2 Blue Margarita

**Drink Category:** Ordinary Drink

**Glass:** Cocktail glass



**Instructions:** Rub rim of cocktail glass with lime juice. Dip rim in coarse salt. Shake tequila, blue curacao, and lime juice with ice, strain into the salt-rimmed glass, and serve.

**Date Modified:** 2015-08-18 14:51:53

#### Drink 3 Tommy's Margarita

**Drink Category:** Ordinary Drink

**Glass:** Old-Fashioned glass



**Instructions:** Shake and strain into a chilled cocktail glass.

**Date Modified:** 2017-09-02 18:37:54

#### Drink 4 Whitecap Margarita

**Drink Category:** Other/Unknown

**Glass:** Margarita/Coupette glass



**Instructions:** Place all ingredients in a blender and blend until smooth. This makes one drink.

**Date Modified:** 2015-09-02 17:00:22

## Question 2: Full-stack MERN Application with CRUD Operations.

Create a MongoDB database of car autosafety information. The name of the database should be **finalexamdb** and collection name should be **autosafety**.

Import the file **finalexamdb.json** in to your collection. The json file contains six documents.

The fields for the collection are: “**carid**”, “**modelname**”, “**brand**”, “**price**”, and “**safetyscore**”. An “**\_id**” field has been created automatically by MongoDB and we can use it to uniquely identify each document.

Build a backend (Node/Express server) and a front-end (React UI). Connect the application to the database created in the previous step and implement CRUD operations.

- Get all the documents from the collection and display them on front-end  
*When the application loads, it should get all the documents from the collection.* (20 Marks)
- Add new data to the collection from the front-end and view the updated data  
*The user should be able to add a new document to the collection.* (20 Marks)
- Delete existing data in the collection through front-end and view the updated data  
*The user should be able to delete the existing document from the collection.* (15 Marks)
- Update existing data in the collection through front-end and view the updated data  
*For the update/edit operation, the user can only update “**price**” and “**safetyscore**”.* (15 Marks)

**Note: If you successfully build the front-end and the backend, but failed to connect with the database, you will lose points.**

The following excerpt shows how the database will look like.

finalexamdb.autosafety

DOCUMENTS 6		TOTAL SIZE 604B	AVG. SIZE 101B	INDEXES 1		TOTAL SIZE 36.0KB	AVG. SIZE 36.0KB
Documents	Aggregations	Schema	Explain Plan	Indexes	Validation		
FILTER							
ADD DATA VIEW							
Displaying documents 1 - 6 of 6							
autosafety							
	_id ObjectId	modelname String	brand String	price Int32	safetyscore Int32	carid Int32	
1	5fd6b271be33cfc0c44e50bf	"Outback"	"Subaru"	30995	21	101	
2	5fd6b294be33cfc0c44e50c0	"Camry"	"Toyota"	23456	18	102	
3	5fd6b2ccbe33cfc0c44e50c1	"Maxima"	"Nissan"	27250	23	103	
4	5fd73b92bb975ac310a0a558	"Sonata"	"Hyundai"	32198	25	104	
5	5fd73be4bb975ac310a0a559	"Optima"	"Kia"	28495	22	105	
6	5fd73c04bb975ac310a0a55a	"Model 3"	"Tesla"	51600	18	106	

The following excerpt shows how the front-end will look like. You are not required to use bootstrap. You can choose your own styles for this question.

Car Safety Scores Information

Add New Record

Car Id	Model Name	Brand Name	Price	Safety Score	Actions	
101	Outback	Subaru	30995	21	Update	Delete
102	Camry	Toyota	23456	19	Update	Delete
103	Maxima	Nissan	27250	23	Update	Delete
104	Sonata	Hyundai	32198	25	Update	Delete
105	Optima	Kia	28495	22	Update	Delete
106	Model 3	Tesla	51600	18	Update	Delete

When user clicks the “Add New Record” Button

Add New Car Details Here..

Car Id

Model Name

Brand

Price

Safety Score

Submit

Home

When user clicks the “Update” Button

Edit the details..

Car Id

Model Name

Brand

Price

28495

Safety Score

51600

Submit

Home