**ITD103 – Modern Query Language**

**Exploring Front-End Development**

**Laboratory Exercises #2**

Name: **Hanan Kiram Ramos**

**Objective:** To familiarize students with the integration process between back-end and front-end systems, specifically focusing on MongoDB as the back-end database.

**Instruction:**

1. Use and enhance the back end developed in lab exercises #2
2. Develop a front-end (CRUD) for your web application.
3. **Follow the instruction in the provided video.** Link - [https://drive.google.com/file/d/1jNdiLkxSbgLC67BPTkrK8Xka9oWPUbo/view?usp=sharing](https://drive.google.com/file/d/1-jNdiLkxSbgLC67BPTkrK8Xka9oWPUbo/view?usp=sharing)

1. Add a **search function** to the web application.

**You can enhance and modify the exercises by adding and incorporating other features you think necessary (additional points will be given)**

**Deliverables:**

1. Create a detailed workflow (step by step) on the exercises**.** **PROVIDED BELOW**
2. Github Repository: **https://github.com/hananramss/Modern-Query-Language-ITD103**
3. Create a Demo Video for your output: **https://drive.google.com/drive/folders/1j6DOelmgdohLjWtySV\_Ok2-5R5oCL8yj**

A screenshot of a computer

Description automatically generatedA screenshot of a website

Description automatically generated

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**FRONTEND WORKFLOW**

To make HTTP requests, **install Axios** using the command: **npm install axios**. Also, **import** other necessary dependencies.

1. **State Variables**: (e.g., const [date, setDate] = useState();)
2. Add a **constant** to handle the submission and addition/updating/deletion of transactions.
3. **Create the form** to add a transaction.

EditTransModal.js

AddTransModal.js

DeleteTransModal.js

**App.js** defines the app's structure, including components, pages, and routes, using *BrowserRouter as Router, Routes, and Route* for seamless navigation.

App.js

**For MODALs**

The **Index.js** is responsible for rendering the root component into the DOM (Document Object Model)

Index.js

Brief Descriptions

JS Files

Folders

Steps

**Legends:**

**Import** these files to the **App.js** to *include* their respective pages in the main application.

Home.js

TransHistory.js

The **Constant.js** file contains all the *API links* required to establish connections between the Backend and Frontend. Utilize these API links with **Axios**.

1. **State Variables**: (e.g., const [transactions, setTransactions] = useState([]);)
2. Add a **constant** to handle the *Search Bar, checkboxes, update, and delete functions.*
3. Create a **useEffect** to fetch transactions from the server to *retrieve data* from the MongoDB Database.
4. **Create a table** with headers and rows to present transaction details. Implement features for checkboxes, updating, and deleting transactions in alignment with design and user interface specifications.

DeleteMultipleTransModal.js

The **Header.js** file contains the code for the *navigation bar.*

The **Styles Folder** contains all the CSS files responsible for styling and designing the web app.

Header.js

Components

Others.js

Constant.js

Modal

Create folders.

Utils

Styles

Pages

Create a frontend file: **Npx create-react-app frontend**