**Task 5**

**Title:** Dynamic table

**Description:** Create dynamic table based on array values it will automatic create table for that.

**Project Overview:**

The project is a React-based application with two dashboard once allows users to input personal data (ID, Name, Age, and City) and dynamically display this data in a table format. The data is stored in the browser's local storage. And other one take the data from Data.js file.

**Key Features:**

* **Dynamic Data Entry**: Users can input their details through a form, which updates the displayed table in real-time.
* **Local Storage Integration**: The application saves user data in the browser's local storage, allowing data retrieval even after the application is closed or refreshed.

**Technical Details:**

* **Framework**: React
* **Hooks Used**: **useState** for state management and **useEffect** for side effects (loading data from local storage).
* **Styling**: The application uses Tailwind CSS for styling, providing a modern and clean interface.
* **Data Management**: Data is managed in a state array and saved to local storage as a JSON string, and pre-defined data stored in Data.js file.

**Code Structure:**

The main component of the application is **Table1.**

* **Imports**: React and necessary components (**DynamicTable** and data).
* **Table1 Component**: Contains the table header and maps through the imported data to render **DynamicTable** components for each data entry.
* **DynamicTable Component**: A reusable component that receives props for individual data entries and displays them in a table row.

The main component of the application is **Table2.**

* **State Variables**:
  + **formData**: Stores the current input values.
  + **dataArray**: Stores the array of submitted user data.
* **Effect Hook**: Loads previously stored data from local storage when the component mounts.

**Example Code Snippet:**

In **Table 1:**

 </div>

       {data.map((card,index)=>(

            <DynamicTable key={index} id={card.id} name={card.name} age={card.age} city={card.city} />

        ))}

       </div>

This code helps to dynamically create more table rows according to how much data stored in Data.js file.

In **Table 2:**

//save data

 const saveData = (e) => {

        e.preventDefault();

        setDataArray([...dataArray,formData])

        localStorage.setItem('formData', JSON.stringify([...dataArray,formData]));

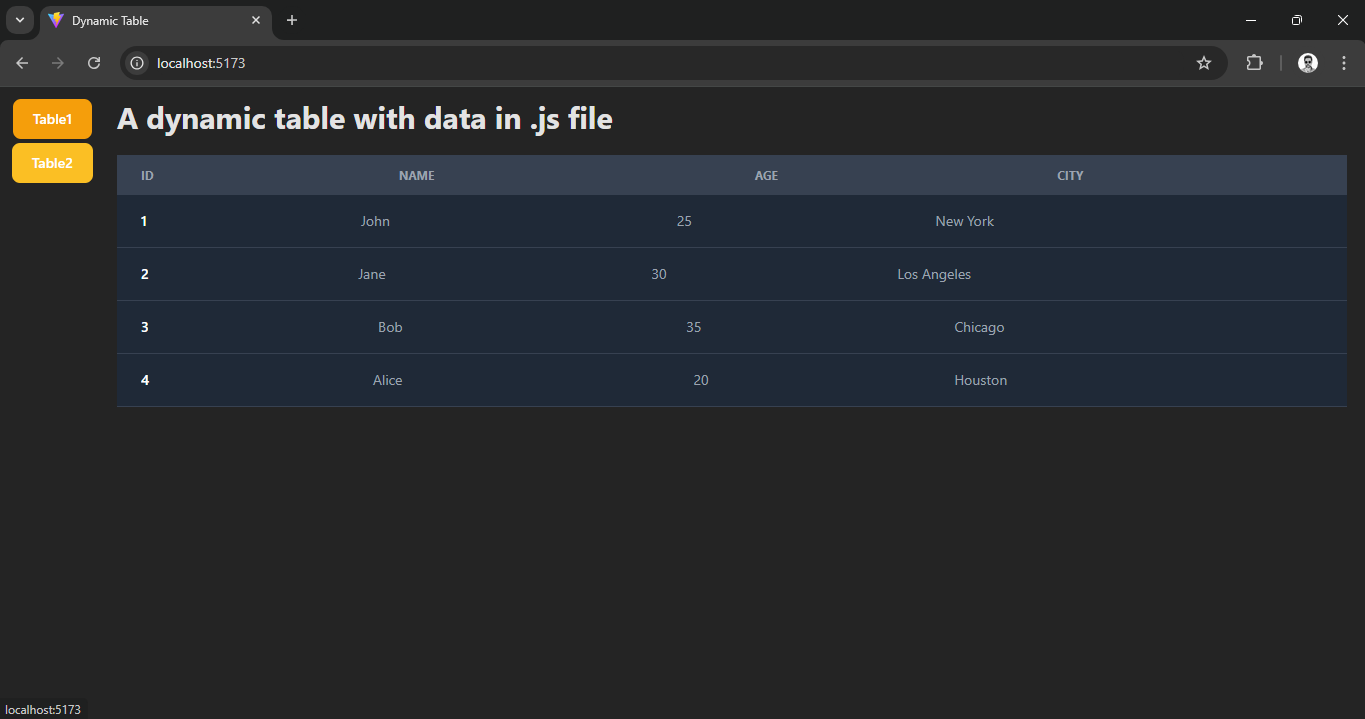
        console.log([...dataArray,formData]);

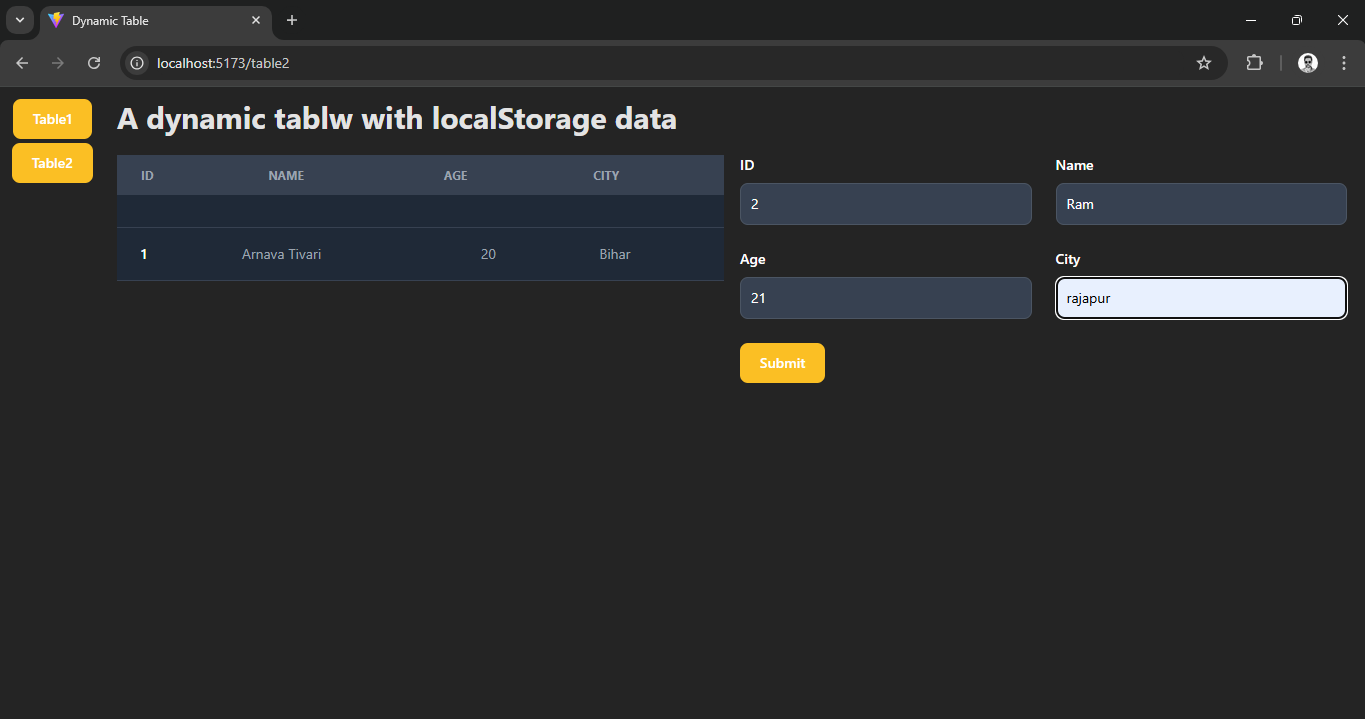
        setFormData({ id: '', name: '', age: '', city: '' });

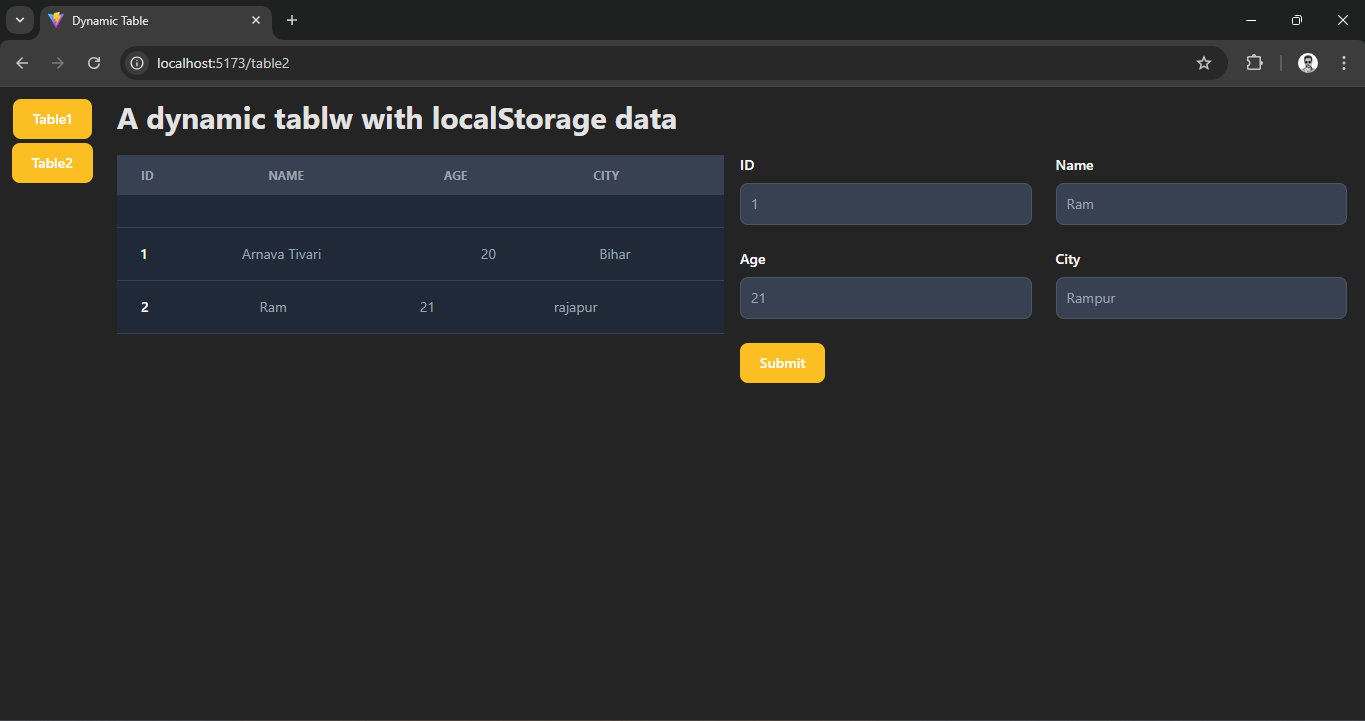
  };

This code helps to store the input data of the form into the LocalStorage of the browser in array format.

**Screenshots:**

****

****

****

**Conclusion:**

This project showcases how to create dynamic table with a functional data entry application using React. By utilizing local storage and pre-defined data in Data.js file.