**Real-Time CRUD Application Documentation**

**Table of Contents**

**1. Introduction**

- Overview

- Technologies Used

**2. Getting Started**

- Prerequisites

- Installation

- Running the Application

**3. Usage Guide**

- Adding New Items

- Editing Existing Items

- Deleting Items

- Searching for Items

**4. Real-Time Data Synchronisation**

- Firebase Integration

- Listening to Real-Time Updates

**5. Code Structure**

- Vue.js Components

**1. Introduction**

**Overview**

The Real-Time CRUD Application is a web-based system built using Vue.js, Firebase Realtime Database, and Tailwind CSS.. Users can perform Create, Read, Update, and Delete (CRUD) operations on the data, and changes are instantly synchronized in real-time with all connected users.

**Technologies Used**

* **Vue.js:** Vue.js is a JavaScript framework for building user interfaces. It forms the core of the application's frontend
* **Firebase Realtime Database:** Firebase Realtime Database is used as the backend data storage solution. It enables real-time data synchronization across users, ensuring that any changes are instantly reflected.
* **Tailwind CSS:** Tailwind CSS is utilized for styling the application. It allows for efficient and customizable styling using utility classes.

**2. Getting Started**

**Prerequisites**

* A modern web browser (e.g., Google Chrome, Mozilla Firefox).
* Internet connectivity.

**Installation**

* Npm install.
* Npm run dev.

**Running the Application**

1. Application will be running locally at => [http://localhost:5173](http://localhost:5173/)/
2. To run the application publically, simply open a web browser and enter the URL <https://2ee2-103-59-207-187.ngrok-free.app/>(I’ll provide the latest URL while testing). The application will load, and users can immediately start interacting with it.

**3. Usage Guide**

**Adding New Items**

1. Click the "Add Item" button.
2. A form will appear, allowing you to enter the name and description of the new item.
3. After entering the details, click the "Add" button to add the item to the dataset.

**Editing Existing Items**

1. Find the item you want to edit in the list.
2. Click the "Edit" button next to the item.
3. The item's details will be displayed in an editable form.
4. Modify the name or description as needed.
5. Click the "Update" button to save your changes.

**Deleting Items**

1. Locate the item you wish to delete in the list.
2. Click the "Delete" button next to the item.
3. A confirmation prompt will appear. Click "OK" to confirm the deletion or "Cancel" to abort.

**Searching Items**

1. Use the search input box located at the top of the page.
2. Enter keywords related to the item you're looking for.
3. As you type, the list of items will filter to display matching results.

**5. Real-Time Data Synchronisation**

**Firebase Integration**

Firebase Realtime Database is integrated into the application as the data storage and synchronization backend. It provides a real-time data exchange mechanism that allows changes made by one user to be instantly reflected for all connected users.

**Listening to Real-Time Updates**

The application uses Firebase's event listener to monitor changes in the dataset. When any user adds, updates, or deletes an item, Firebase triggers events, and the Vue.js components responsible for displaying the data automatically update to reflect the changes. This real-time behavior ensures that all users see the most current data without needing to refresh the page.

**7. Code Structure**

**Vue.js Components**

The codebase is organized using Vue.js components to maintain a structured and modular architecture. Key components include:

* **Data Table:** Responsible for displaying the list of items, including the table structure and the client-side search functionality.
* **Add/Edit Item Form:** Handles the form for adding new items and editing existing ones.
* **Firebase Configuration:** Manages the Firebase configuration and integration.