Note & task management system

Introduction, purpose, objectives and technologies

In the fast-paced world of personal and professional management, staying organized is crucial for productivity and efficiency. Recognizing this need, the Note & Task Management App was developed as a solution to help users manage their tasks and notes effectively. This web application is designed to streamline the process of tracking and organizing tasks while providing a robust system for managing associated notes.

Purpose and Objectives

The primary purpose of the Note & Task Management App is to provide users with a user-friendly platform for managing their daily tasks and related notes. The app aims to address common challenges associated with task management, such as:

- **Task Organization**: Helping users organize their tasks into a manageable list, where they can easily view, add, edit, and delete tasks.
- **Note Integration**: Allowing users to add detailed notes to each task, ensuring that important details and progress updates are not overlooked.
- **User Authentication**: Providing secure access to tasks and notes through user authentication, ensuring that each user's data remains private and accessible only to them.

The app's objectives include:

- **Enhancing Productivity**: By offering a centralized place for task and note management, the app aims to enhance user productivity and organization.
- **Providing Real-Time Updates**: Utilizing Firebase Firestore, the app ensures that task and note updates are reflected in real time, providing users with the latest information at all times.
- **Ensuring Security and Privacy**: Implementing Firebase Authentication to securely manage user accounts and ensure that each user’s data is protected.

Technologies Used

The Note & Task Management App uses modern web technologies to deliver a seamless and efficient user experience:

- **Vue.js**: A progressive JavaScript framework used to build the app's interactive and dynamic user interface. Vue.js is chosen for its flexibility and ease of integration with other libraries and tools.
- **Firebase**: A comprehensive backend platform that provides authentication, real-time database, and hosting services. Firebase Authentication manages user sign-up and sign-in processes, while Firebase Firestore serves as the NoSQL database for storing and retrieving tasks and notes.

Features of the app

Overview

The app is a Vue.js-based Note & Task Management Application that integrates with Firebase for user authentication and Firestore for storing tasks and notes. It allows users to manage their tasks and related notes, including the ability to add, edit, mark as done, and delete both tasks and notes. Users sign in or sign up via Firebase authentication and can only see and manage their own data, securely stored under their user ID.

Features of the application

1. **User Authentication:**
    - Users can sign up, sign in, and sign out using Firebase authentication.
    - After logging in, the app ensures that the user remains authenticated while they manage their tasks and notes.
    - A sign-out button is provided to allow users to securely log out.
2. **Task Management:**
    - Tasks are displayed in a scrollable sidebar on the left side of the app.
    - Users can add new tasks, edit existing tasks, and delete tasks.
    - Tasks can be marked as “done,” visually indicating their completion.
    - Long task titles are truncated for a clean display.
    - When a task is selected, it is highlighted, and related notes are displayed in the main content area.
    - Task deletion also removes all associated notes from the Firestore database.
3. **Note Management:**
    - For each task, users can add multiple notes. Notes are managed in the main content area, which changes based on the selected task.
    - Notes can be added, edited, deleted, and marked as done.
    - Similar to tasks, there are buttons for editing and deleting notes.
    - When a note is edited, the form appears directly at the note’s location (inline editing).
    - Notes are tied to the task and the authenticated user, ensuring that only the relevant notes are displayed for each user.
4. **Firestore Integration:**
    - Tasks and notes are stored in Firestore under the user’s specific path. This ensures that each user’s data is private and securely stored.
    - Operations for tasks and notes such as add, edit, delete, and fetching are handled via Firebase Firestore queries.
    - Batch operations are used to delete tasks along with their associated notes.
5. **Responsive Layout:**
    - The app features a clean and responsive layout. The sidebar displays the tasks, while the main content area shows the notes for the selected task.
    - Both the sidebar (tasks) and the main content area (notes) are scrollable, allowing users to navigate through a large number of items comfortably.

How the App Works:

1. **User logs in:** Upon successful login, Firebase's authentication listener updates the app state, and tasks for the authenticated user are fetched from Firestore.
2. **Task Selection:** When a task is selected, the app displays its title and loads all associated notes in the main content area.
3. **Adding Tasks and Notes:** Users can create tasks from the sidebar. After selecting a task, they can add notes from the main content area.
4. **Editing and Deleting:** Tasks and notes can be edited or deleted. Changes are reflected in real-time by fetching data from Firestore after any update.
5. **Task and Note Completion:** Tasks and notes can be marked as "done," visually updating their status.
6. **User logs out:** When the user signs out, the authentication state is cleared, and the app redirects to the login page.

This app is a functional tool for managing personal tasks and notes, offering features like real-time updates with Firestore, user authentication, and task/note management.

How things work?

1. Firebase Setup

The app is connected to Firebase for two main purposes:

- Authentication
- Cloud Firestore (NoSQL database)

1.1 Firebase Authentication:

Firebase Authentication handles user registration, sign-in, and sign-out. This ensures that users can securely access their tasks and notes. Firebase Auth is set up to track the authenticated user via the auth.onAuthStateChanged method, which updates the app state when a user logs in or logs out.

User Authentication Flow:

- When a user signs in, Firebase Auth assigns a unique uid (user ID) to each user.
- The app checks for authentication state changes using Firebase's auth.onAuthStateChanged() method. When a user is logged in, the app fetches the user's tasks and notes using this uid.
- Upon logging out, the app resets the state and redirects the user to the login page.

1.2 Cloud Firestore (Database):

Firestore is the backend NoSQL database where tasks and notes are stored and retrieved. Each user has their own set of tasks and notes stored under their unique uid in Firestore.

- **Firestore Structure:** users/{userId}/tasks/{taskId}/notes/{noteId}

    - **users**: Top-level collection that contains documents with userId as the document ID.
    - **tasks**: Sub-collection under each user that contains task documents with taskId as the document ID.
    - **notes**: Sub-collection under each task, storing related notes with noteId as the document ID.

Each task document contains:

- title (String): The title of the task.
- isDone (Boolean): Whether the task is marked as done.
- createdAt (Timestamp): The creation date of the task.

Each note document contains:

- content (String): The content of the note.
- isDone (Boolean): Whether the note is marked as done.
- createdAt (Timestamp): The creation date of the note.

2. Vue.js Frontend and Firebase Integration

The frontend of the app is built using Vue.js, which interacts with Firebase to provide a seamless experience for task and note management. Here's how different parts of the Vue.js app are connected and communicate with Firebase.

2.1 Authentication (Vue.js + Firebase Auth)

- **Sign-up and Login**: When a user signs up or logs in, Firebase Auth generates a uid and maintains the user's session. The Vue.js app checks if a user is logged in using the Firebase Auth observer (auth.onAuthStateChanged()), which allows the app to reactively manage the authenticated user's state and trigger necessary actions like fetching tasks.
- **Sign-out**: The signOut() method of Firebase Auth is used to log out the user. This clears the session and redirects the user to the login page.

2.2 Firestore CRUD Operations (Vue.js + Firebase Firestore)

- **Fetching Tasks**:
    - When a user logs in or performs specific actions (like adding/deleting a note or task), the app queries Firestore to fetch the tasks for the authenticated user using their uid. This is done through getDocs() to retrieve all tasks belonging to the logged-in user.
- **Adding a Task**:
    - When a user adds a new task, a addDoc() call is made to the tasks sub-collection under the user's document in Firestore. The new task document is then automatically synced with Firestore.
- **Editing a Task**:
    - When a task is edited, the app uses updateDoc() to modify the title or isDone field of a specific task document in Firestore.

- **Deleting a Task**:
    - When a task is deleted, both the task document and its related notes are deleted. A deleteDoc() operation is performed on the taskId, and a batch operation is used to delete all notes associated with the task.
- **Fetching Notes**:
    - Notes for a specific task are fetched via getDocs() from the notes sub-collection under the specific task document. The task's taskId is used to fetch only notes associated with that task.
- **Adding a Note**:
    - Users can add notes to specific tasks. The app calls addDoc() to insert a new note under the selected task's notes sub-collection in Firestore.
- **Editing a Note**:
    - Notes can be edited by updating their content using updateDoc() on the specific noteId in Firestore.
- **Deleting a Note**:
    - The app uses deleteDoc() to remove a note from the Firestore under the specific taskId and noteId.
- **Marking Tasks and Notes as Done**:
    - Both tasks and notes have an isDone field that can be toggled between true and false. When users mark a task or note as done, the app updates the corresponding document using updateDoc() and modifies the isDone field.

3. Vue.js Components and Data Flow

3.1 Task Component

The Task.vue component handles:

- Displaying the list of tasks in a scrollable sidebar.
- Allowing users to add, delete, edit, and mark tasks as done.
- Emitting events to the parent component (App.vue) when a task is selected, added, or deleted.

3.2 Notes Component

The Notes.vue component handles:

- Displaying the notes for a selected task in the main content area.
- Allowing users to add, delete, edit, and mark notes as done.
- Fetching notes dynamically when a task is selected.
- Handling inline editing for notes directly at their location.
- Syncing updates to Firestore when changes are made.

3.3 App Component

The App.vue component is the main controller that:

- Manages the state of the user, tasks, and notes.

- Handles user authentication and routing based on login state.
- Displays the sidebar with tasks and the main content with notes for the selected task.

3.4 Event Handling

- The Task component emits events to App.vue when a task is selected, deleted, or added. This ensures the App.vue component knows which task to show in the Notes.vue component.
- Similarly, the Notes component emits events when notes are added or modified, so the app can refresh the task list when necessary.

4. UI and Layout

- **Responsive Design**: The app uses basic responsive design principles, with a scrollable sidebar for tasks and a flexible main content area for notes. The UI adjusts well for different screen sizes and ensures ease of use.
- **Real-Time Updates**: Although the app doesn't use Firestore's real-time listeners, data is fetched and updated dynamically whenever tasks or notes are added, edited, or deleted, ensuring a fresh display of content.

Project Conclusion

The Note & Task Management App is a comprehensive web application designed to help users efficiently manage their tasks and associated notes. The app leverages Firebase for backend services, including authentication, real-time data storage, and database management. Built with Vue.js for a dynamic and interactive front-end, the app provides a user-friendly interface for managing tasks and notes.

How to Access the App

The Note & Task Management App is deployed and accessible online. You can visit the app using the following URL:

Note & Task Manager App

Instructions:

1. Open the provided URL in your web browser.
2. If you are not logged in, you will need to sign up or log in using your email address to access the app's features.
3. Once logged in, you can start managing your tasks and notes through the intuitive interface.