



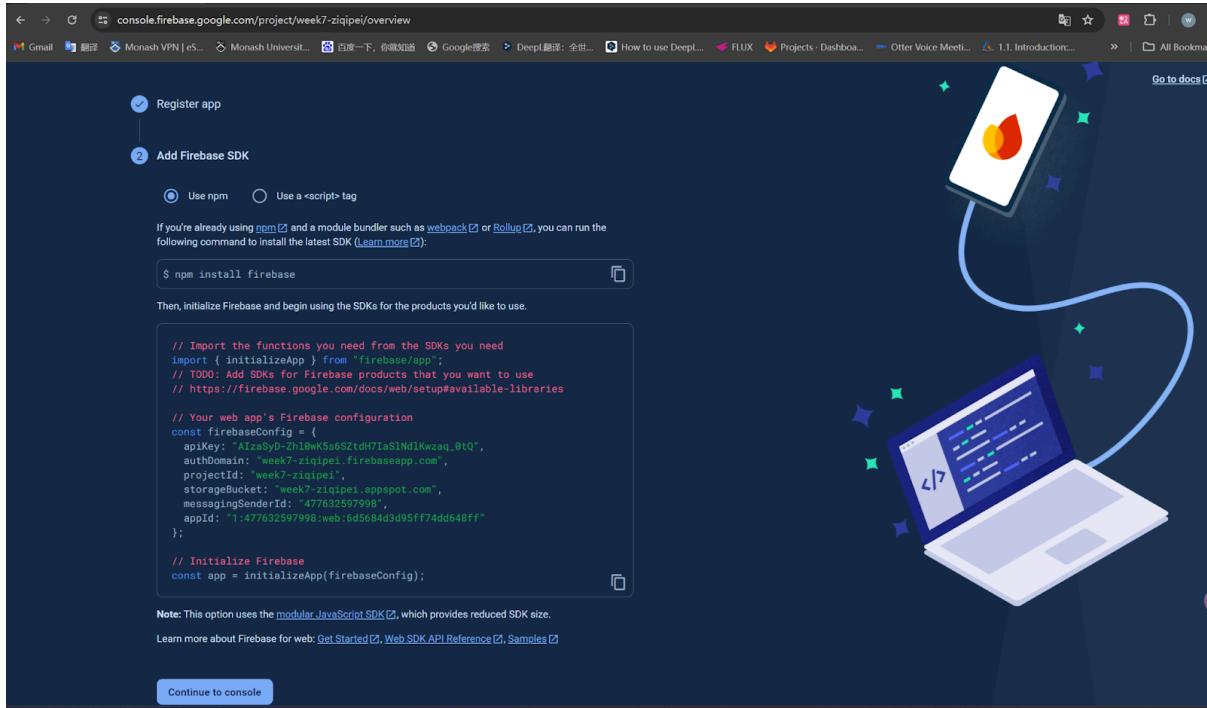
eFolio activity 7&8

Ziqi Pei 33429472 Lab Thursday 10am-12am Github rep:
https://github.com/Franz-Pei/Franz/tree/881769a859b603ccc0f8cac91d9818b70a5c24e5/MONASH_IT_MASTER/FIT5032Internet_applications_development

7.1. Lab set up

A screenshot of a web browser showing the Firebase console. The URL in the address bar is "console.firebaseio.google.com/project/week7-ziqipei/overview". The page title is "Week7-ZiqiPei". The main content area features a dark background with two 3D avatars of people. One is pointing towards a yellow and red flame-like icon. Below them is the text "Get started by adding Firebase to your app" and a "Spark plan" button. At the bottom, there's a callout box for "Gemini in Firebase" with a "Try it now" button. The browser's sidebar shows other projects like "Gmail", "翻译", "Monash VPN | eS...", "Monash University...", "百度一下，你就知道", "Google搜索", "Deep翻译: 全世...", "How to use DeepL...", "FLUX", "Projects - Dashboard", "Otter Voice Meet...", "1.1. Introduction...", and "All Bookmarks".

7.1.2 Create web app in Firebase project



send email

The screenshot shows the Firebase Authentication settings page. The 'Sign-in method' tab is selected. It displays the 'Email/Password' provider status as 'Enabled'. There is also a section for 'SMS Multi-factor Authentication' with a note about enabling it for advanced features. At the bottom, there is a 'Upgrade to enable' button.

7.2. VueJS setup

```

<template>
  <div class="container">
    <header class="d-flex justify-content-center py-3">
      <ul class="nav nav-pills">
        <li class="nav-item">
          <router-link to="/" class="nav-link active-class="active">Home</router-link>
        </li>
        <li class="nav-item">
          <router-link to="/about" class="nav-link active-class="active">About</router-link>
        </li>
        <li v-if="loginStatus" class="nav-item">
          <button @click="performLogout" class="btn btn-link nav-link">Logout</button>
        </li>
        <li class="nav-item">
          <router-link to="/firelogin" class="nav-link active-class="active">Firebase Login</router-link>
        </li>
      </ul>
    </header>
  </div>
</template>
<script>
import { mapState, mapActions } from 'vuex';
</script>
<style>
</style>
<script>
export default {
  data(){
    DEBUG CONSOLE TERMINAL PORTS
    PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\FIT5032-week7> npm install firebase
    added 85 packages in 13s
    3 packages are looking for funding
    run 'npm fund' for details
    PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\FIT5032-week7> []
    > OPPO_桌面 苏奶奶的小熊: 实时翻译打开 [隐藏]
    > TUIP_桌面 苏奶奶的小熊: 实时翻译打开 [隐藏]
    > PROBLEMS
    Filter (e.g. text, **"/ts, tsconfig.json")
    No problems have been detected in the workspace.
  }
}

```

Copy the code Firebase generated when creating web app to New fire

```

<template>
  <div class="container">
    <header class="d-flex justify-content-center py-3">
      <ul class="nav nav-pills">
        <li class="nav-item">
          <router-link to="/" class="nav-link active-class="active">Home</router-link>
        </li>
        <li class="nav-item">
          <router-link to="/about" class="nav-link active-class="active">About</router-link>
        </li>
        <li v-if="loginStatus" class="nav-item">
          <button @click="performLogout" class="btn btn-link nav-link">Logout</button>
        </li>
        <li class="nav-item">
          <router-link to="/firelogin" class="nav-link active-class="active">Firebase Login</router-link>
        </li>
      </ul>
    </header>
  </div>
</template>
<script>
import 'bootstrap/dist/css/bootstrap.min.css'
</script>
<style>
</style>
<script>
import PrimeVue from 'primevue/config'
import Aura from '@primevue/themes/aura'
// src/Firebase/Firebase.js
import { initializeApp } from 'firebase/app';
const app = createApp(App)
app.use(PrimeVue, {
  theme: {
    preset: Aura
  }
});
const firebaseConfig = {
  apiKey: "AIzaSyD-Zh10wK5s6Sztdh7Tia1Nd1Kwzad_0tQ",
  authDomain: "week7-zipei.firebaseioapp.com",
  projectId: "week7-zipei",
  storageBucket: "week7-zipei.appspot.com",
  messagingSenderId: "477632597998",
  appId: "1:477632597998:web:6d5684d5d95ff74dd648ff"
};
initializeApp(firebaseConfig);
app.use(store);
app.use(router);
app.mount('#app')
</script>
DEBUG CONSOLE TERMINAL PORTS
3 packages are looking for funding
run 'npm fund' for details
PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\FIT5032-week7> npm install firebase
added 85 packages in 13s
3 packages are looking for funding
run 'npm fund' for details
    > OPPO_桌面 苏奶奶的小熊: 实时翻译打开 [隐藏]
    > TUIP_桌面 苏奶奶的小熊: 实时翻译打开 [隐藏]
    > PROBLEMS
    Filter (e.g. text, **"/ts, tsconfig.json")
    No problems have been detected in the workspace.

```

7.3. Registration Process

```

<template>
  <p></p>
  <input type="password" placeholder="Password" v-model="password" />
  <p></p>
  <button @click="register">Save to Firebase</button>
</template>

<script setup>
import { ref } from 'vue';
import { getAuth, createUserWithEmailAndPassword } from 'firebase/auth';
import { useRouter } from 'vue-router';

const email = ref("");
const password = ref("");
const router = useRouter();
const auth = getAuth();

const register = () => {
  createUserWithEmailAndPassword(auth, email.value, password.value)
    .then((userCredential) => { // userCredential is the correct term here
      console.log("Firebase Register Successful!");
      console.log(userCredential.user); // Accessing the User object
      router.push("/firlogin");
    })
    .catch((error) => {
      console.error("Error during registration:", error.message); // More detailed error message
    });
}
</script>

```

DEBUG CONSOLE TERMINAL PORTS

```

3 packages are looking for funding
run 'npm fund' for details
PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\fit5032-week7> npm install firebase
added 85 packages in 13s
added 85 packages in 13s

3 packages are looking for funding
run 'npm fund' for details
PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\fit5032-week7>

```

7.4 Sign In Process

```

<template>
  <input type="password" placeholder="Password" v-model="password" />
  <p></p>
  <button @click="signin">Sign in via Firebase</button>
</template>

<script setup>
import { ref } from 'vue';
import { getAuth, signInWithEmailAndPassword } from 'firebase/auth';
import { useRouter } from 'vue-router';

const email = ref("");
const password = ref("");
const router = useRouter();
const auth = getAuth();

const signin = () => {
  signInWithEmailAndPassword(auth, email.value, password.value)
    .then(() => {
      console.log("Firebase Register Successful!");
      setTimeout(() => {
        console.log("Current User:", auth.currentUser);
        router.push("/");
      }, 1000); // Delaying the check by 1 second
    })
    .catch((error) => {
      console.error("Error during sign-in:", error.code);
    });
}
</script>

```

DEBUG CONSOLE TERMINAL PORTS

```

3 packages are looking for funding
run 'npm fund' for details
PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\fit5032-week7> npm install firebase
added 85 packages in 13s
added 85 packages in 13s

3 packages are looking for funding
run 'npm fund' for details
PS D:\Monash\GitHub\Franz-Pei\MONASH_IT_MASTER\FIT5032Internet_applications_development\fit5032-week7>

```

To verify the registration.

The screenshot shows a developer's environment with multiple tabs open. The main browser tab displays an authentication dashboard with a warning about cross-origin redirect sign-in support. Below the dashboard is a table of user data. The code editor on the right contains a Vue.js component for Firebase registration, utilizing the `createUserWithEmailAndPassword` method. The bottom of the screen features a terminal window showing a successful Vite build and a list of open files.

```
zpei-library > src > views > FirebaseRegisterView.vue ...  
1  <template>  
2    <div>  
3      <h1>Create an Account</h1>  
4      <input type="text" placeholder="Email" v-model="email" />  
5      <input type="password" placeholder="Password" v-model="password" />  
6      <button @click="register">Register</button>  
7    </div>  
8  </template>  
9  
10 <script setup>  
11   import { ref } from 'vue';  
12   import { getAuth, createUserWithEmailAndPassword } from 'firebase/auth';  
13   import { useRouter } from 'vue-router';  
14  
15   const email = ref('');  
16   const password = ref('');  
17   const router = useRouter();  
18   const auth = getAuth(); // Get the authentication instance  
19  
20   const register = () => {  
21     createUserWithEmailAndPassword(auth, email.value, password.value)  
22       .then((userCredential) => {  
23         console.log("Firebase Register Successful!", userCredential);  
24         router.push('/FireLogin'); // Redirect after successful  
25       })  
26       .catch((error) => {  
27         console.error("Error during registration:", error.message);  
28       });  
29   };  
30 </script>  
31
```

To verify whether user is signed in. You can go to the Console panel in the browser to see what has been printed

The screenshot shows a browser window with the following details:

- Title Bar:** localhost:5173/FirebaseLogin
- Header:** Home About Login **Firebase Login** Firebase Register
- Content:** Sign In form with Email, Password, and Sign In buttons.
- Developer Tools:** Opened in the bottom-left corner, showing:
 - Elements tab: Shows the DOM structure of the sign-in page.
 - Console tab: Shows log messages, including "Translate: Running handler for translators" and "Firebase Sign-in Successful!".
 - Sources tab: Shows the source code for "FirebaseSignInView.vue".
- Explorer Tab:** Shows the project structure with files like "node_modules", "public", "src", "assets", "components", "index.js", "store.js", "views", and "README.md".
- Bottom Right:** Shows tabs for "DEBUG CONSOLE", "TERMINAL", and "PORTS".

set different user with admin



```
(4) (+0000006): Translate: Parsing code for Library Catalog (Aleph) (cf7ecab-041d-b954-795a-208d348999d5; 2021-05-21 04:15:00) debug.js
(4) (+0000006): Translate: Parsing code for unAPI (e701c1ac-1e37-ad4b-b678-a8e834b0e08; 2019-07-15 19:40:00) debug.js
(4) (+0000006): Translate: Parsing code for COINS (05d07af9-105a-4572-99f6-a8e231c0def4; 2021-06-03 14:25:00) debug.js
(4) (+0000001): Translate: Parsing code for Embedded Metadata (091c027d-74ac-47d4-a107-9c3069ab7b48; 2024-06-14 04:50:00) debug.js
(3) (+0000001): Translate: Embedded Metadata; found 2 meta tags. debug.js
(4) (+0000001): Translate: Parsing code for DOI (159dcfe8a3-4301-a499-30fe5409c340d; 2024-08-21 22:15:00) debug.js
(3) (+0000001): Translate: All translators detect calls and RPC calls complete: debug.js
(3) (+0000000): No suitable translators found debug.js
(5) (+0000000): Translate: Running handler 0 for translators debug.js

Firebase sign-in successful! Firebase sign-in view state
+ userImpl: {providerId: "firebase", proactiveRefresh: reloadListenerInfo: {}, reloadListener: null, uid: "SWYKKMaahQg06GnxzKq5jgxT2x", ...} Firebase sign-in view state
  accessToken: "eyJhbGk101JSU1IiwiU2hpdD1yAytTAwzE2ZmRkOAT0ZTvnG00D0ExWzNMiLzGzI0tgotK00E1C10oXAI0V1Qf0jQ...eyJpcMI01oDhwCzov3N1Y3VzXvBavZuLmdvbdzsZ55jb20vd2vLaZctemlxaxBlas1ImI2C16IndlZws3LXppwWzWk1iLChHxRo3RpHml0JE3M"
  auth: AutImpl {app: FirebaseAppImpl, heartbeatServiceProvider: Provider, appCheckServiceProvider: Provider, config: {}, currentUser: _UserImpl, ...}
  displayName: null
  email: "spei000@student.monash.edu"
  emailVerified: false
  isAnonymous: false
  metadata: {metadata {createdAt: "1726147896876", lastSignInAt: "172630763509", lastSignInTime: "Fri, 13 Sep 2024 12:32:43 GMT", creationTime: "Thu, 12 Sep 2024 13:31:36 GMT"}}

+ metadata: {metadata {createdAt: "1726147896876", lastSignInAt: "172630763509", lastSignInTime: "Fri, 13 Sep 2024 12:32:43 GMT", creationTime: "Thu, 12 Sep 2024 13:31:36 GMT"}}

+ phoneNumbers: []
+ photoURL: null
+ proactiveRefresh: ProactiveRefresh {user: _UserImpl, isRunning: true, timerId: 319, errorBackoff: 30000}
+ providerData: []
+ providerId: "firebase"
+ reloadListener: null
+ refreshTokenInfo: {localId: "SWYKKMaahQg06GnxzKq5jgxT2x", email: "spei000@student.monash.edu", passwordHash: "UKVEQUMHJRQ-", emailVerified: false, passwordUpdatedAt: 1726147896876, ...}
+ stokenManager: StokenManager {refreshToken: "AMF-V8xt-Y5zJstpqhyCmzRu84jL1Vxsns0f4R6jD2Vov2LToVxK1ld0Ajgrzyame-obRe9x5fC1Sc960H1kt9yJhsfw", accessToken: "eyJhbGk101JSU1IiwiU2hpdD1yAytTAwzE2ZmRkOAT0ZTvnG00D0ExWzNMiLzGzI0tgotK00E1C10oXAI0V1Qf0jQ...eyJpcMI01oDhwCzov3N1Y3VzXvBavZuLmdvbdzsZ55jb20vd2vLaZctemlxaxBlas1ImI2C16IndlZws3LXppwWzWk1iLChHxRo3RpHml0JE3M", uid: "SWYKKMaahQg06GnxzKq5jgxT2x", ...}
```

8.2 VueJS setup

The screenshot shows a developer's workspace with the following components:

- Code Editor:** The left pane displays the `firebase.js` file from the `src/firebase` directory. The code initializes a Firebase app with specific configuration details.
- Terminal:** The bottom-left shows a terminal window with the command `npm install firebase` and its output, indicating 85 packages were added.
- File Explorer:** The right pane is the "EXPLORER" view, showing the project structure for "FIT5032-WEEK7". It includes folders for `node_modules`, `zpei.library`, `vscode`, `src`, and `public`. The `src` folder contains `assets`, `components`, and `firebase`. Within `firebase`, the `index.js` file is selected.

8.4 Use Firestore to create new data in library system

```

19 <script>
20 import { ref } from 'vue';
21 import db from './firebase/init.js'
22 import {collection, addDoc} from 'firebase/firestore';
23
24 import BookList from '../components/BookList.vue';
25
26 export default{
27   setup(){
28     const isbn = ref('');
29     const name = ref('');
30
31     const addBook = async () =>{
32       try{
33         const isbnNumber = Number(isbn.value);
34         if(isNaN(isbnNumber)){
35           alert('ISBN must be a valid number');
36           return;
37         }
38
39         await addDoc(collection(db, 'books'),{
40           isbn: isbnNumber,
41           name: name.value
42         });
43         isbn.value = '';
44         name.value = '';
45         alert('Book added successfully !');
46       } catch(error){
47         console.error('Error adding book: ', error);
48       }
49     };
50
51     return{
52       addBook
53     }
54   }
55 }

```

Port 5174 is in use, trying another one...

VITE v5.3.5 ready in 495 ms

- Local: <http://localhost:5175/>
- Network: use --host to expose
- Vue DevTools: Open http://localhost:5175/_devtools/ as a separate window
- Vue DevTools: Press Alt+Shift+F0 in App to toggle the Vue DevTools

→ press h + enter to show help

All done, after running your program and adding the book. The Firestore database will reflect the data that has been posted

Cloud Fire... Add database

Multiple databases now available Now you can create and manage more than one database in the same project

Data Rules Indexes Usage Extensions

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing Configure App Check

Panel view Query builder

books > 9yTtVfhlgSQVxjiNLGxW

(default)	books	9yTtVfhlgSQVxjiNLGxW
+ Start collection	+ Add document	+ Start collection
books >	9yTtVfhlgSQVxjiNLGxW >	+ Add field
	DSSbuJH0FLkbwS...	isbn: 1012
	KJ0H47Hooist0B...	name: "Study vue"
	KLcvxt5WRVZGCB...	
	SLUthj5D2yf46...	
	WGuqX8heoNoZty...	
	Wniat94cwM7jJ6...	
	mTHeFASgMxin10...	

15 import { collection, query, where, getDocs } from 'firebase/firestore';
16
17 export default {
18 setup() {
19 const books = ref([]);
20
21 const fetchBooks = async () => {
22 try {
23 const q = query(collection(db, 'books'), where('name', '==', 'Study vue'));
24 const querySnapshot = await getDocs(q);
25 const booksArray = [];
26
27 querySnapshot.forEach((doc) => {
28 booksArray.push({ id: doc.id, ...doc.data() });
29 });
30
31 books.value = booksArray;
32 } catch (error) { // corrected 'catch' to 'catch' here
33 console.error('Error fetching books: ', error);
34 }
35 };
36
37 onMounted(() => {
38 console.log('Component is mounted');
39 fetchBooks().then(() => {
40 console.log('Books fetched:', books.value);
41 });
42 });
43
44 return {
45 books
46 };
47 }
48 }

VITE v5.3.5 ready in 528 ms

- Local: <http://localhost:5173/>
- Network: use --host to expose
- Vue DevTools: Open http://localhost:5173/_devtools/ as a separate window
- Vue DevTools: Press Alt+Shift+F0 in App to toggle the Vue DevTools

→ press h + enter to show help

8.5 Use query to find data from Firestore

The screenshot shows a Vue.js application interface. At the top, there is a navigation bar with links: Home, About, Add Book (highlighted in blue), Login, and Firebase Login. Below the navigation bar is a "Firebase Register" button.

The main content area has a title "Add Book". Below it is a form with two input fields: "ISBN:" and "Name:", both currently empty. A "Add Book" button is positioned below the form. To the right of the form, a list titled "Book with ISBN > 1000" displays two items:

- Study vue - ISBN: 1012
- Ziqi Pei - ISBN: 10000

On the far right, the VS Code sidebar is visible, showing the project structure and files. The file "main.ts" is open in the editor. The sidebar includes sections for "zpeii-library", "src", "views", and "problems", with a message indicating "No problems have been detected in the workspace."

```
2   <div>
3     <book-list>
4   </div>
5 </template>
6
7 <script>
8   import { ref } from 'vue';
9   import db from './firebase/init.js';
10  import { collection, addDoc } from 'firebase/firestore';
11
12  import BookList from '../components/BookList.vue';
13
14  export default{
15    setup(){
16      const isbn = ref('');
17      const name = ref('');
18
19      const addBook = async () =>{
20        try{
21          const isbnNumber = Number(isbn.value);
22          if(isNaN(isbnNumber)){
23            alert('ISBN must be a valid number');
24            return;
25          }
26
27          await addDoc(collection(db, 'books'),{
28            isbn: isbnNumber,
29            name: name.value
30          });
31          isbn.value = '';
32          name.value = '';
33          alert('Book added successfully !');
34        } catch(error){
35          console.error('Error adding book: ', error);
36        }
37
38      }
39
40      return{
41        isbn,
42        name,
43        addBook
44      };
45    },
46    components: {
47      BookList
48    }
49  };
50
51  <!-- This component is located in the views directory -->
52 <AboutView>
53 <AddBookView>
54 <FirebaseRegisterView>
55 <FirebaseSignInView>
56 <HomeView>
57 <LoginView>
58 <App.vue>
59 <main.js>
60 # style.css
61 # eslintrc.js
62 # gitignore
63 # .nocheckercss
64 > OUTLINE
65 > TIMELINE
66 > PROBLEMS
67
68 Filter (e.g. text, **.ts, !**.no...)
```

Week8HD

update data

The screenshot shows a Vue.js application interface. On the left, there's a sidebar with a tree-like navigation structure:

- components
- icons
- ▀ BHeader.vue
- ▀ BookList.vue U
- ▀ JSON.vue
- ▀ firebase
- ▀ init.js
- ▀ router
- ▀ index.js
- ▀ store
- ▀ store.js
- ▀ views
- ▀ AboutView.vue
- ▀ AddBookView.vue
- ▀ FirebaseRegisterView.vue
- ▀ FirebaseSignInView.vue
- ▀ HomeView.vue
- ▀ LoginView.vue
- ▀ App.vue
- ▀ main.js
- # style.css
- ▀ eslintrc.js

The main content area has a title "Add Book" and a form to add a new book entry. Below it, there's a table titled "Book with ISBN > 1000" showing two entries.

```
books.value = querySnapshot.docs.map(doc => {
  const data = doc.data();
  return {
    id: doc.id,
    name: data.name || 'Unnamed', // Default to 'Unnamed' if name
    isbn: data.isbn || 0, // Fetch the 'isbn' field, default to 0
  };
});
console.log("Books loaded:", books.value);
};

// Update book's name and ISBN in Firestore
const updateBook = async (bookId, newName, newISBN) => {
  const bookRef = doc(db, 'books', bookId);
  try {
    const docSnap = await getDoc(bookRef);
    if (docSnap.exists()) {
      const currentDate = docSnap.data();
      // Update both name and ISBN
      await updateDoc(bookRef, [
        name: newName || currentDate.name, // Update name
        isbn: newISBN ? Number(newISBN) : currentDate.isbn // Update ISBN
      ]);
      console.log("Document updated with ID:", bookId);
      await fetchBooks(); // Refresh the list after update
    } else {
    }
  } catch (err) {
    console.error("Error updating document", err);
  }
};
```

Study vue1	12000	Save	Delete
Study vue22	130000	Save	Delete

The screenshot shows the Google Cloud Firestore console interface. At the top, there's a navigation bar with tabs: Data (selected), Rules, Indexes, Usage, and Extensions. Below the navigation bar is a banner with a shield icon and the text "Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing" followed by a "Configure App Check" button. On the right side of the banner is a close button (X).

The main area displays a hierarchical document structure under the path "books > FN8qjGSTaA3sz...".

- (default)**: A collection named "books" is shown. It contains two documents:
 - 6HBYxgW3vGhx0kHZaCZB
 - FN8qjGSTaA3szIxPvi71** (selected): This document has fields: "isbn: 130000" and "name: 'Study vue22'".

On the right side of the interface, there are buttons for "Panel view" and "Query builder", and a three-dot menu icon. At the bottom right, there's a "More in Google Cloud" link.

where The data

```

setup() {
    const books = ref([]);

    // Fetch books with ISBN > 1000, ordered by ISBN, limited to 5 results
    const fetchBooks = async () => {
        const q = query(
            collection(db, 'books'),
            where('isbn', '>', 1000), // Filter by ISBN > 1000
            orderBy('isbn', 'desc'), // Order by ISBN in descending order
            limit(5) // Limit the results to 5
        );
        const querySnapshot = await getDocs(q);
        books.value = querySnapshot.docs.map(doc => {
            const data = doc.data();
            return {
                id: doc.id,
                name: data.name || 'Unnamed', // Default to 'Unnamed' if none
                isbn: data.isbn || 0, // Fetch the 'isbn' field, or 0 if none
            };
        });
        console.log("Books loaded:", books.value);
    };
}

```

Order by

```

// Update book's name and ISBN in Firestore
const updateBook = async (bookId, newName, newISBN) => {
    const bookRef = doc(db, 'books', bookId);
    try {
        const docSnap = await getDoc(bookRef);
        if (docSnap.exists()) {
            const currentData = docSnap.data();
            // Update both name and ISBN
            await updateDoc(bookRef, {
                name: newName || currentData.name, // Update name
                isbn: newISBN ? Number(newISBN) : currentData.isbn // Update ISBN
            });
            console.log("Document updated with ID:", bookId);
            await fetchBooks(); // Refresh the list after update
        } else {
            console.log("No such document!");
        }
    } catch (error) {
        console.error("Error updating document: ", error);
    }
};

// Delete book from Firestore
const deleteBook = async (bookId) => {
    const bookRef = doc(db, 'books', bookId);
    try {
        await deleteDoc(bookRef);
        console.log("Document deleted with ID:", bookId);
        await fetchBooks(); // Refresh the list after deletion
    } catch (error) {
        console.error("Error deleting document: ", error);
    }
};

```

Order by

The screenshot shows a development environment with two main panes. The left pane displays the source code for a Vue.js component, specifically a list of books from a Firebase database. The right pane shows a table of books with ISBNs greater than 1000, with buttons for adding new books and deleting existing ones.

```
<script>
import { ref, onMounted } from 'vue';
import db from '../firebase/init.js';
import { collection, query, where, getDocs, orderBy, limit, doc, updateDoc, deleteDoc } from 'firebase/firestore';
import { collectionGroup } from 'firebase/firestore';

export default {
  setup() {
    const books = ref([]);

    // Fetch books with ISBN > 1000, ordered by ISBN, limited to 5 results
    const fetchBooks = async () => {
      const q = query(
        collection(db, 'books'),
        where('isbn', '>', 1000), // Filter by ISBN > 1000
        orderBy('isbn', 'desc'), // Order by ISBN in descending order
        limit(5) // Limit the results to 5
      );
      const querySnapshot = await getDocs(q);
      books.value = querySnapshot.docs.map(doc => {
        const data = doc.data();
        return {
          id: doc.id,
          name: data.name || 'Unnamed', // Default to 'Unnamed' if no name
          isbn: data.isbn || 0, // Fetch the 'isbn' field, or 0 if none
        };
      });
      console.log("Books loaded:", books.value);
    };

    // Update book's name and ISBN in Firestore
    const updateBook = async (bookId, newName, newISBN) => {
      const bookRef = doc(db, 'books', bookId);
      await updateDoc(bookRef, {
        name: newName,
        isbn: newISBN
      });
    }
  }
};

// Fetch books with ISBN > 1000
const fetchBooks = async () => {
  const q = query(
    collection(db, 'books'),
    where('isbn', '>', 1000), // Filter by ISBN > 1000
    orderBy('isbn', 'desc'), // Order by ISBN in descending order
    limit(5) // Limit the results to 5
  );
  const querySnapshot = await getDocs(q);
  books.value = querySnapshot.docs.map(doc => {
    const data = doc.data();
    return {
      id: doc.id,
      name: data.name || 'Unnamed', // Default to 'Unnamed' if no name
      isbn: data.isbn || 0, // Fetch the 'isbn' field, or 0 if none
    };
  });
  console.log("Books loaded:", books.value);
};

// Update book's name and ISBN in Firestore
const updateBook = async (bookId, newName, newISBN) => {
  const bookRef = doc(db, 'books', bookId);
  await updateDoc(bookRef, {
    name: newName,
    isbn: newISBN
  });
}

// Add a new book
const addBook = (name, isbn) => {
  const newBookRef = doc(db, 'books');
  const newBookData = {
    name: name,
    isbn: isbn
  };
  await setDoc(newBookRef, newBookData);
};

// Delete a book
const deleteBook = (bookId) => {
  const bookRef = doc(db, 'books', bookId);
  await deleteDoc(bookRef);
};


```

ISBN	Name	Action
130000	Study vue22	Save Delete
120444	I love study fit5032	Save Delete
120443	w5-ziqi	Save Delete
12000	Study vue1	Save Delete
1012	Ziqi Pei	Save Delete

limit vue page 5(add new data so if the vue is gather than 1012 the data will instand by 1013)

