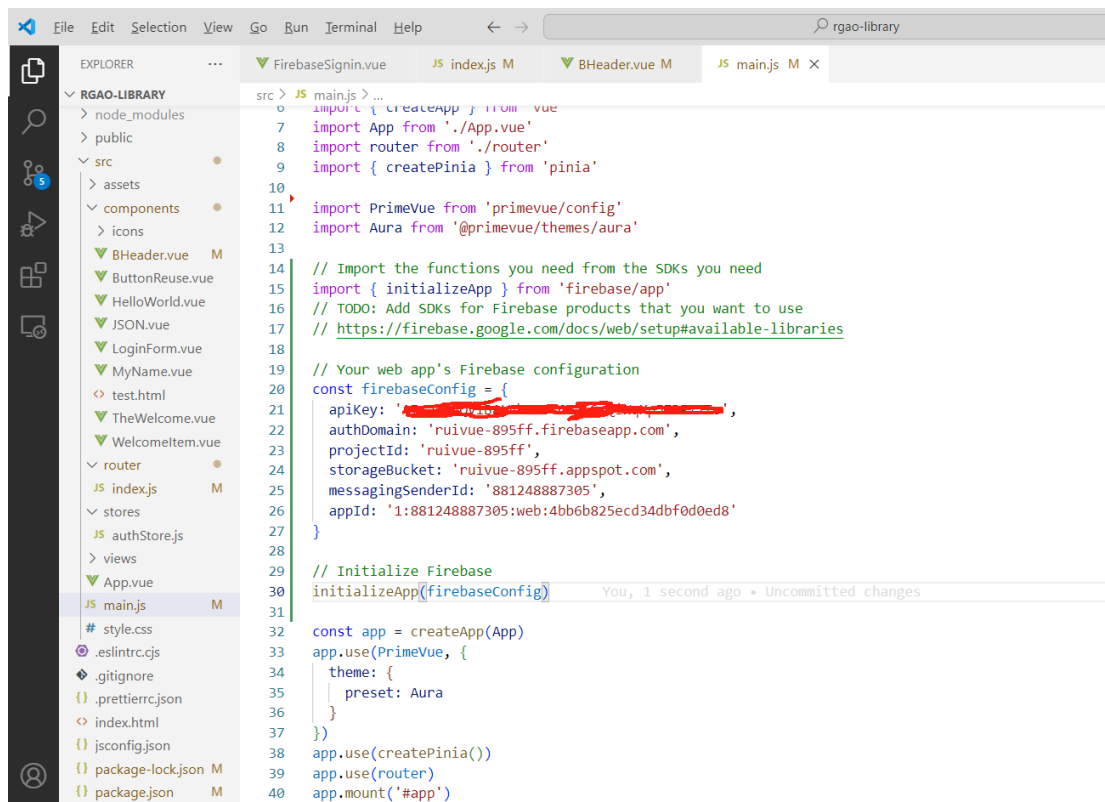
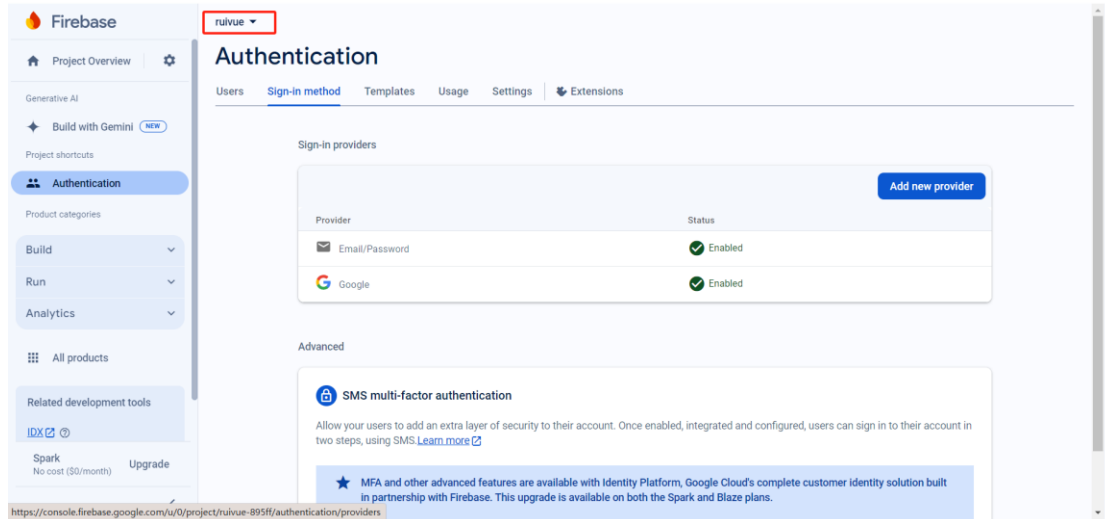


# Task 7.1

Create project & Register your project with Firebase:



Register user via Firebase:

Monash

单词学习网站

英语电视节目

英语听力

老师推荐网站

学习

Gmail

Online Java Comp...

All Bookmark

Firebase

Project Overview

Generative AI

Build with Gemini

Project shortcuts

Authentication

Product categories

Build

Run

Analytics

All products

Authentication

Users

Sign-in method

Templates

Usage

Settings

Extensions

Cross-origin redirect sign in on Google Chrome M115+ is no longer supported and will stop working on 24 June 2024.

Search by email address, phone number or user UID

Add user

Identifier	Providers	Created	Signed in	User UID
rgao123@gmail.com		3 Sept 2024	3 Sept 2024	GewoYjxx0bQSwMs076Q6CUS...

Rows per page: 50 | 1 - 1 of 1

Monash

单词学习网站

英语电视节目

英语听力

老师推荐网站

学习

Gmail

Online Java Comp...

Home (Week 5)

FireLogin

FireRegister

About

Create an Account

Elements

Console

Network

Application

Sources

Performance

Memory

Security

Lighthouse

Recorder

Performance insights

top

Filter

auth/email-already-in-use

File

Edit

Selection

View

Go

Run

Terminal

Help

rgao-library

Vue

JS

Vue

Vue

src > views > FirebaseRegister.vue > {} script setup

```
1 <template>
2   <h1>Create an Account</h1>
3   <p><input type="text" placeholder="Email" v-model="email"></p>
4   <p><input type="text" placeholder="Password" v-model="password"></p>
5   <p><button @click="register">Save to Firebase</button></p>
6
7 </template>
8 <script setup>
9
10 import { ref } from "vue"
11 import { useRouter } from "vue-router"
12 import { getAuth, createUserWithEmailAndPassword } from "firebase/auth"
13 const email = ref("")
14 const password = ref("")
15 const auth = getAuth()
16 const router = useRouter()
17 const register = () => {
18   createUserWithEmailAndPassword(auth, email.value, password.value)
19     .then((data) => {
20       console.log(data)
21       router.push("/FireLogin")
22     }).catch((error) => {
23       console.log(error.code)
24     })
25 }
26 </script>
```

## Singin in via registerd account via Firebase:

The screenshot displays a web application running on localhost:5173. The page features a "User Information Form" with fields for Username, Gender, Password, and Confirm Password. Below the form, the browser's developer console is open, showing the following log entries:

```
proactiveRefresh: ProactiveRefresh (user: _UserImp1, isRunning: false, timerId: null, errorCallback: 30000)
providerData: [{}]
providerId: "firebase"
reloadListener: null
reloadUserInfo: {
  email: "rgao123@gmail.com"
  emailVerified: false
  lastSignIn: "1725320918465"
  lastRefreshAt: "2024-09-02T23:48:40.465Z"
  localId: "GmY3vaBhQ5uFu076Q6CUPJ3Tg1"
  passwordHash: "UNIQUE_HASH"
  passwordUpdatedAt: "1725320492277"
  providerUserInfo: [{}]
}
[[Prototype]]: Object
stateTokenManager: {refreshToken: "AHF-v8yFuZHL1ZG2wSRXfF8Ajjj4VTXK-1EPJfQG3T_1gm7-AzhKpGGu4CponlgJHJ061G1012G418wQdce#183F0aXE-Ts", accessToken: "ey3h86c101750z13h1t4TetpZC16tah3MAAMTgOHm1ZDUyZ", tenantId: null, uid: "GmY3vaBhQ5uFu076Q6CUPJ3Tg1", refreshToken: (...)}
```

# Task 7.2

## Login via general use

[Home \(Week 5\)](#) [FireLogin](#) [FireRegister](#) [Add Book](#) [Edit Book](#) [About](#)

### Sign in

Email address  
user@gmail.com

Email address  
\*\*\*\*\*

Role:  
user

Sign in via Firebase

localhost:5173 says  
You are login as user

OK

Monash 单词学习网站 英语电视节目 英语听力 老师推荐网站 All Book

Home (Week 5) Book About

### Sign in

Email address  
user@gmail.com

Email address  
\*\*\*\*\*

Role:  
user

Sign in via Firebase

Monash 单词学习网站 英语电视节目 英语听力 老师推荐网站 All Bookmarks

Home (Week 5) Book About

## Login via admin role

[Home \(Week 5\)](#) [FireLogin](#) [FireRegister](#) [Add Book](#) [Edit Book](#) [About](#)

### Sign in

Email address  
admin@gmail.com

Email address  
\*\*\*\*\*

Role:  
admin

Sign in via Firebase

localhost:5173 says  
You are login as admin

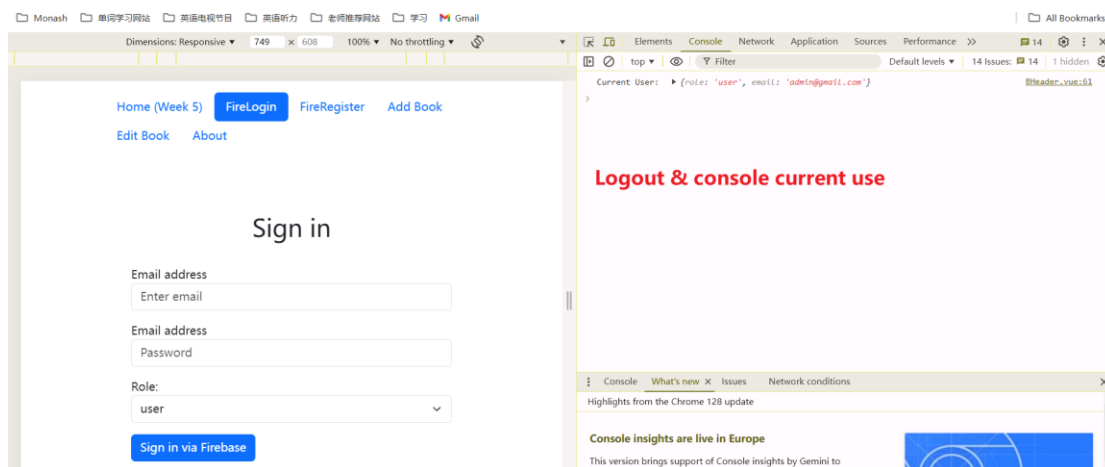
OK

EditBookView.vue UAddBookView.vue UFirebaseSignIn.vue M XJS index.js MJS main.js M

rgao-library > src > views > FirebaseSignIn.vue > {} script setup

```
31 <script setup>
47
48 Codeium: Refactor | Explain | Generate Function Comment | X
49 const register = () => {
50   console.log(email.value, password.value, role.value)
51   signInWithEmailAndPassword(auth, email.value, password.value)
52     .then(async (userCredential) => {
53     const user = userCredential.user
54
55     // get user data from Firestore
56     const userDoc = await getDoc(doc(db, 'users', user.uid))
57     if (userDoc.exists()) {
58       const userData = userDoc.data()
59       switch (role.value) {
60         case 'admin':
61           if (userData.role !== 'admin')
62             throw new Error("Sorry, You are not an admin")
63           alert("You are login as admin")
64           break
65         case 'user':
66           alert("You are login as user")
67           break
68         default:
69           throw new Error("Unknown role: " + userData.role)
70       }
71       userData.role = role.value
72       userStore.saveUser(userData)
73     }
74     console.log(auth.currentUser)
75     router.push("/")
76   }).catch((error) => {
77     console.log(error)
78     if(error.code === 'auth/email-already-in-use') {
79       alert('Email already in use')
80     } else {
81       alert(error.message)
82     }
83   })
84 }
```

Multi Role Login



# Task 8.1

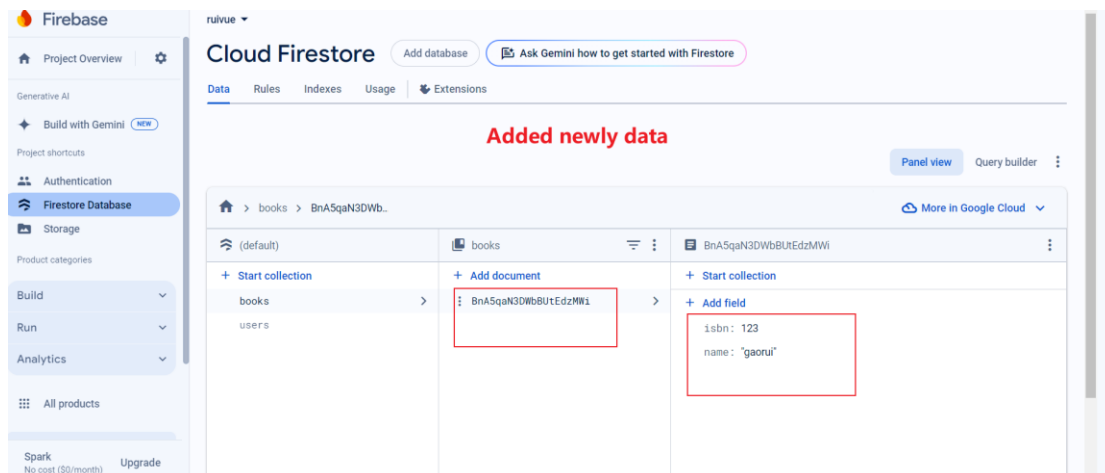
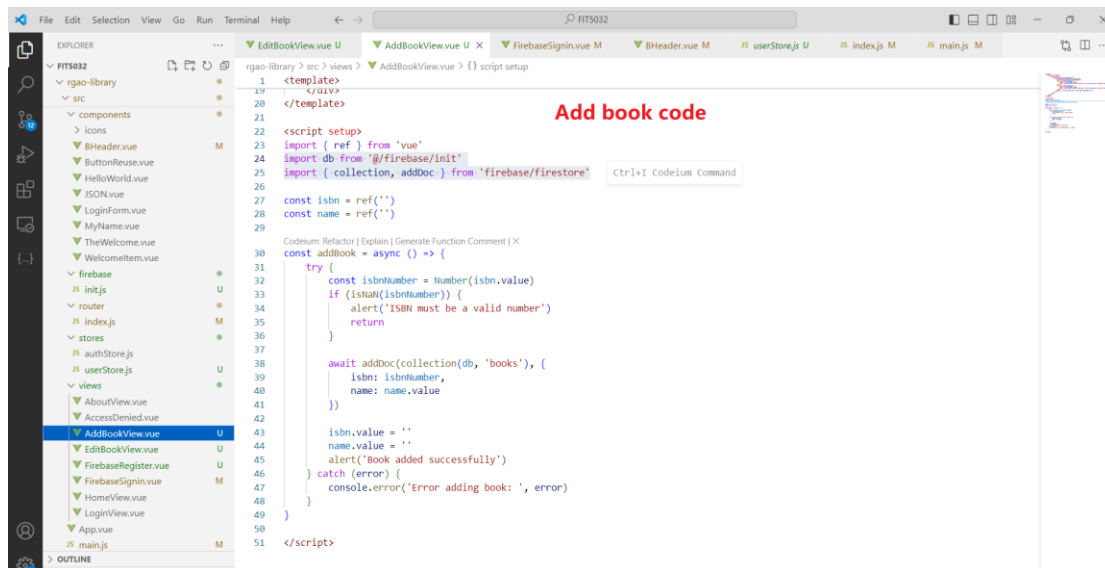


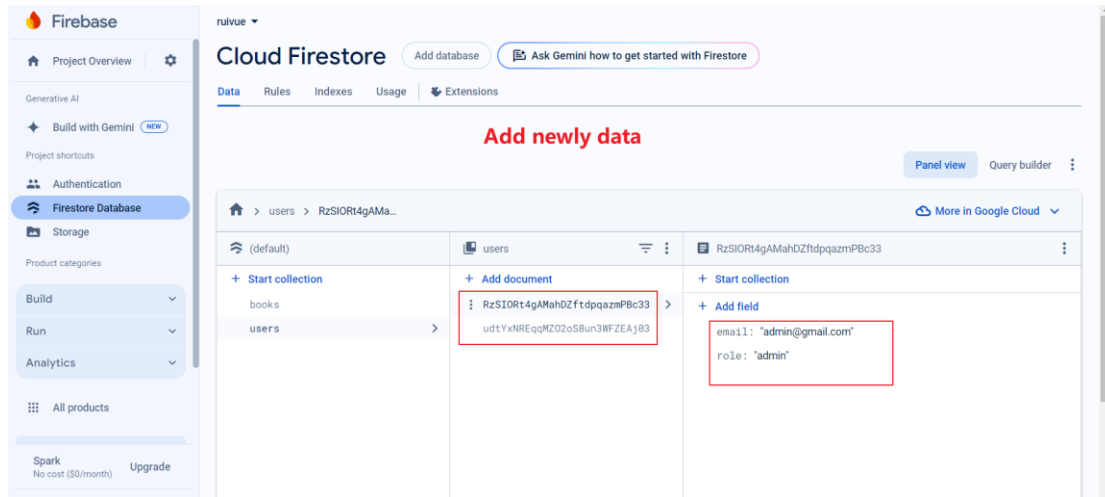
Add book

Add Book

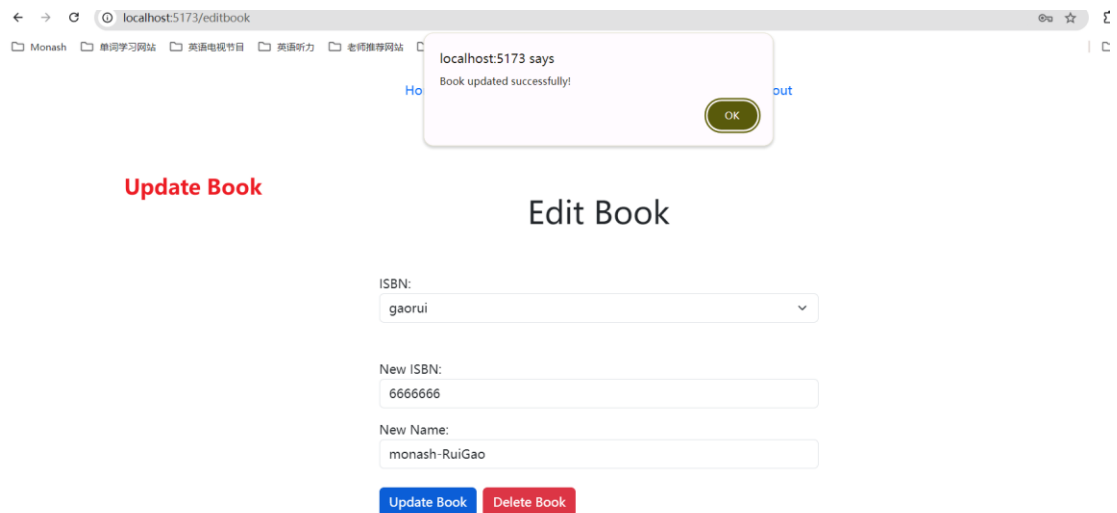
ISBN:

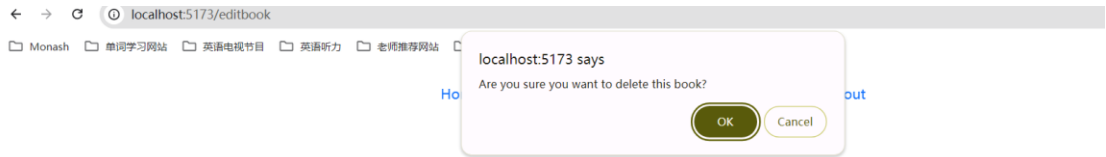
Name:





## Task 8.2





Delete book

## Edit Book

ISBN:  
monash-RuiGao

New ISBN:  
6666666

New Name:  
monash-RuiGao

[Update Book](#) [Delete Book](#)



Delete book

## Edit Book

ISBN:  
monash-RuiGao

New ISBN:  
6666666

New Name:  
monash-RuiGao

[Update Book](#) [Delete Book](#)

[Home \(Week 5\)](#) [Add Book](#) [Edit Book](#) [About](#) [Logout](#)

Book has been deleted

## Edit Book

ISBN:  
Please select a book

New ISBN:  
eg. 978-3-16-148410-0

New Name:  
eg. The Catcher in the Rye

[Update Book](#) [Delete Book](#)



```

<script setup>
import { onMounted, ref, watch } from 'vue';
import { getDocs, collection, doc, updateDoc, deleteDoc } from 'firebase/firestore';
import db from '@/firebase/init';

const selectedIsbn = ref(""); // Store the currently selected book's ISBN
const newIsbn = ref(""); // Store the new ISBN entered by the user
const newName = ref(""); // Store the new name entered by the user
const books = ref([]); // Store the list of books fetched from Firestore

// Fetch all books from Firestore and populate the books array
Codeium: Refactor | Explain | ✕
const fetchBooks = async () => {
  const querySnapshot = await getDocs(collection(db, 'books'));
  books.value = querySnapshot.docs.map(doc => ({ ...doc.data(), id: doc.id }));
};

// Watch for changes in the selected ISBN and automatically populate the form fields
watch(selectedIsbn, (newIsbnValue) => {
  const selectedBook = books.value.find(b => b.isbn === newIsbnValue);
  if (selectedBook) {
    newIsbn.value = selectedBook.isbn; // Set the new ISBN to the selected book's ISBN
    newName.value = selectedBook.name; // Set the new name to the selected book's name
  }
});

// Update the selected book with the new ISBN and/or name
Codeium: Refactor | Explain | ✕
const updateBook = async () => {
  if (selectedIsbn.value && (newIsbn.value || newName.value)) {
    const book = books.value.find(b => b.isbn === selectedIsbn.value);
    const bookRef = doc(db, 'books', book.id);

    // Create an object with updated fields
    const updatedData = {};
    if (newIsbn.value) updatedData.isbn = newIsbn.value;
    if (newName.value) updatedData.name = newName.value;

    await updateDoc(bookRef, updatedData);
    alert('Book updated successfully!');
    fetchBooks(); // Reload books to reflect changes
  } else {
    alert('Please select a book and enter new information to update.');
```

**update book**

```

  }
};

// Delete the selected book from Firestore
Codeium: Refactor | Explain | ✕
const deleteBook = async () => {
  if (selectedIsbn.value && confirm('Are you sure you want to delete this book?')) {
    const book = books.value.find(b => b.isbn === selectedIsbn.value);
    const bookRef = doc(db, 'books', book.id);

    await deleteDoc(bookRef);
    alert('Book deleted successfully!');
    fetchBooks(); // Reload books to reflect deletion
    selectedIsbn.value = ""; // Clear the selection
    newIsbn.value = ""; // Clear the form fields
    newName.value = ""; // Clear the form fields
  } else {
    alert('Please select a book to delete.');
```

**delete book**

```

  }
};

// Fetch books on component mount
onMounted(() => {
  fetchBooks();
});
</script>

```

```

15 <script setup>
16 import { onMounted, ref, watch } from 'vue';
17 import { getDocs, collection, doc, updateDoc, deleteDoc, query, orderBy } from 'firebase/firestore';
18 import db from '@/firebase/init';
19
20 const selectedIsbn = ref(""); // Store the currently selected book's ISBN
21 const newIsbn = ref(""); // Store the new ISBN entered by the user
22 const newName = ref(""); // Store the new name entered by the user
23 const books = ref([]); // Store the list of books fetched from Firestore
24
25 // Fetch all books from Firestore and populate the books array
26 const fetchBooks = async () => {
27   const q = query(collection(db, 'books'), orderBy('name'));
28   const querySnapshot = await getDocs(q);
29   books.value = querySnapshot.docs.map(doc => ({ ...doc.data(), id: doc.id }));
30 };
31
32

```

**Retrive data**