Efolio Week 9 & Week 10

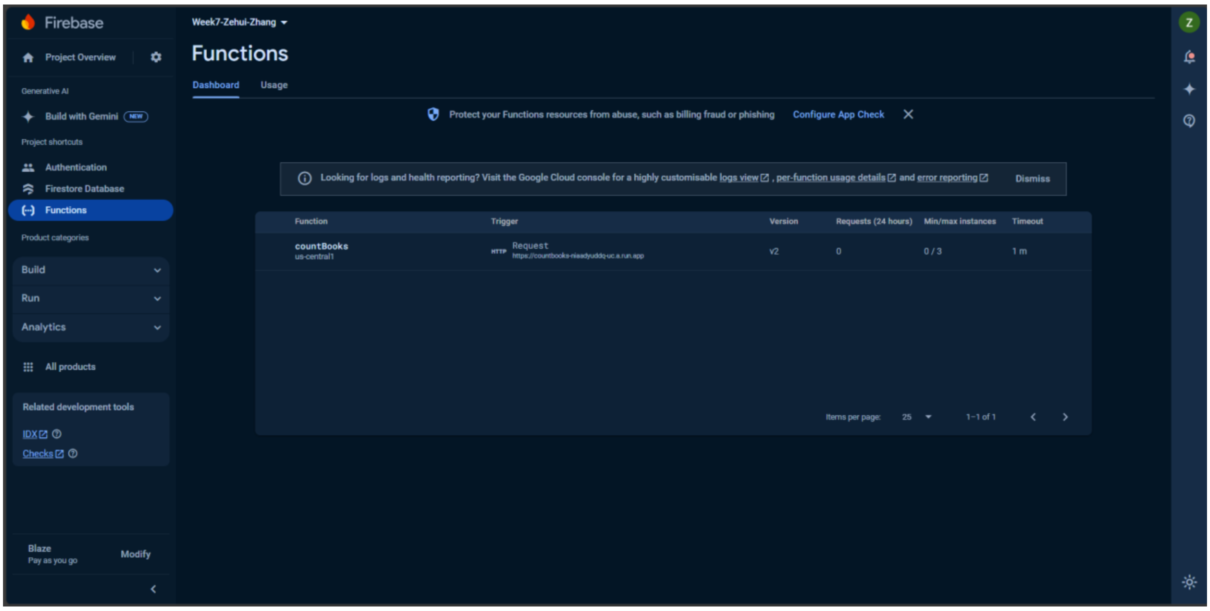
Zehui Zhang

34103600

Efolio week 9

Task 9.1

Screenshot set 1:



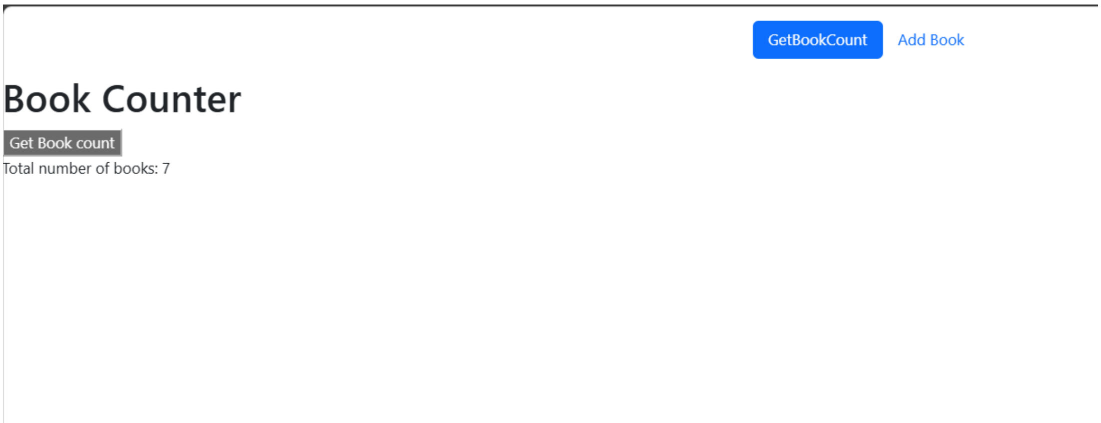
Functions

Protect your Functions resources from abuse, such as billing fraud or phishing

Looking for logs and health reporting? Visit the Google Cloud console for a highly customisable logs view , per-function usage details and error reporting

Function	Trigger	Version	Requests (24 hours)	Min/max instances	Timeout
countBooks us-central1	HTTP Request https://countbooks-niaadyuddq-uc.a.run.app	v2	0	0 / 3	1 m

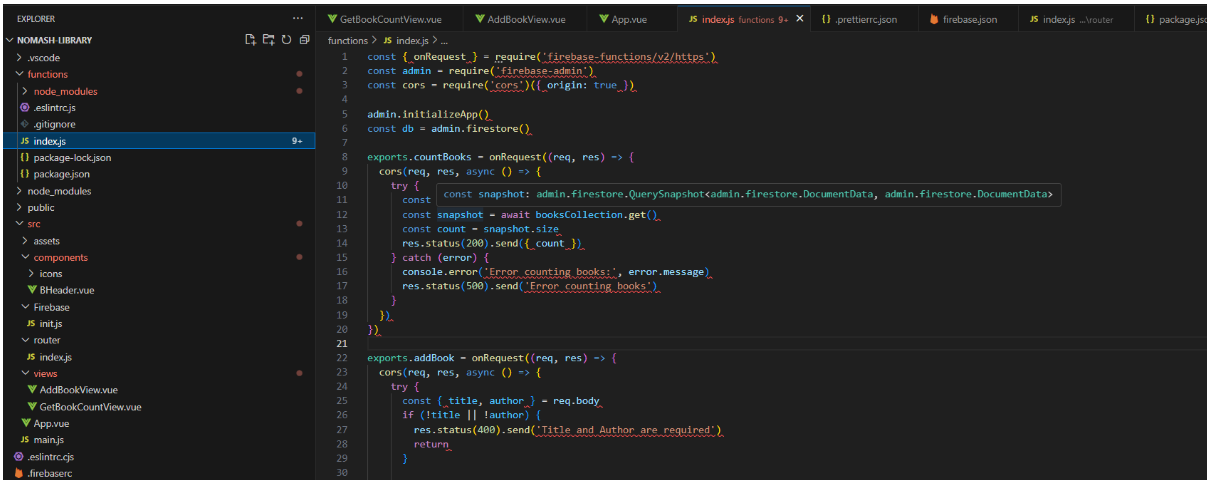
Screenshot set 2:



Book Counter

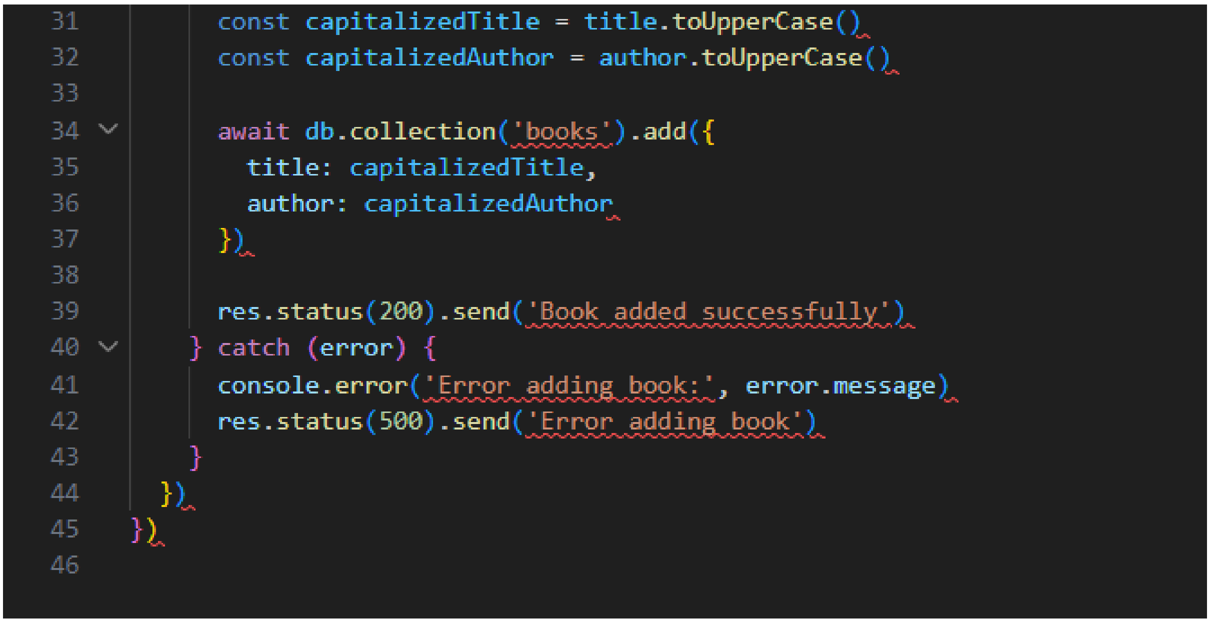
Total number of books: 7

The **index.js** file in the **functions** folder:



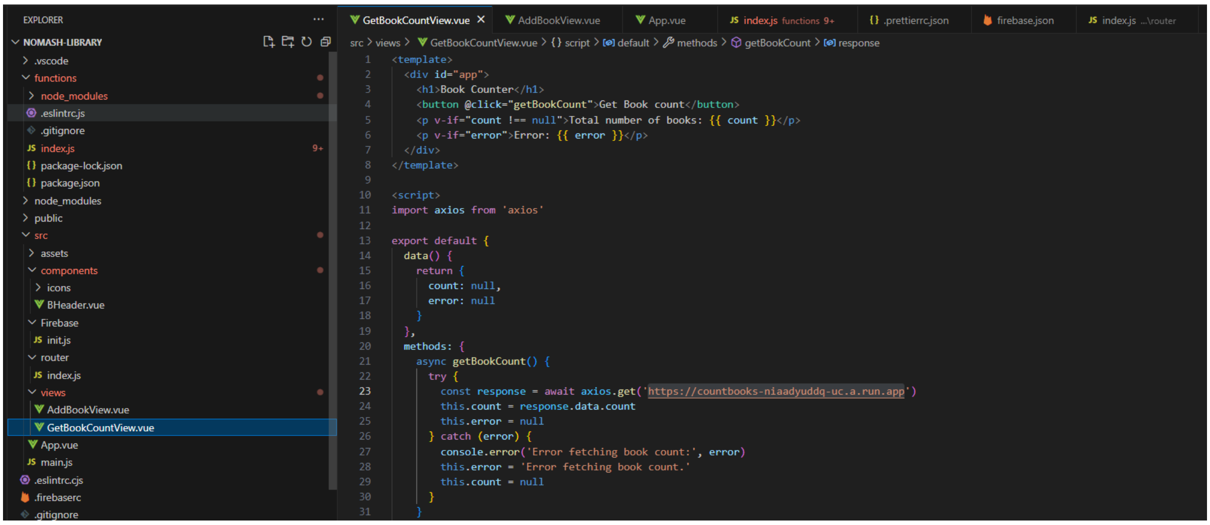
```
const snapshot: admin.firestore.QuerySnapshot<admin.firestore.DocumentData, admin.firestore.DocumentData>
```

```javascript
const { onRequest } = require('firebase-functions/v2/https')
const admin = require('firebase-admin')
const cors = require('cors')({ origin: true })

admin.initializeApp()
const db = admin.firestore()

exports.countBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const
      const snapshot = await booksCollection.get()
      const count = snapshot.size
      res.status(200).send({ count })
    } catch (error) {
      console.error('Error counting books:', error.message)
      res.status(500).send('Error counting books')
    }
  })
})

exports.addBook = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const { title, author } = req.body
      if (!title || !author) {
        res.status(400).send('Title and Author are required')
        return
      }

```

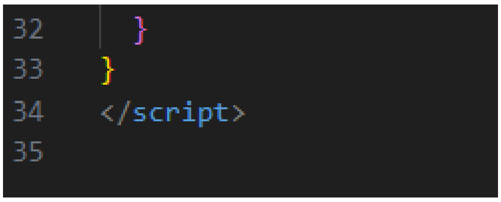


```javascript
      const capitalizedTitle = title.toUpperCase()
      const capitalizedAuthor = author.toUpperCase()

      await db.collection('books').add({
        title: capitalizedTitle,
        author: capitalizedAuthor
      })

      res.status(200).send('Book added successfully')
    } catch (error) {
      console.error('Error adding book:', error.message)
      res.status(500).send('Error adding book')
    }
  })
})

```

GetBookCountView.vue:

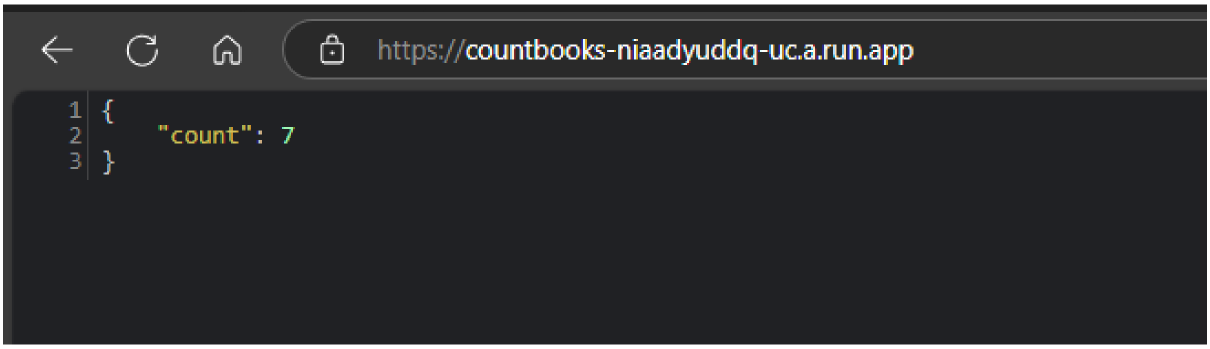


```
<template>
  <div id="app">
    <h1>Book Counter</h1>
    <button @click="getBookCount">Get Book count</button>
    <p v-if="count !== null">Total number of books: {{ count }}</p>
    <p v-if="error">Error: {{ error }}</p>
  </div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      count: null,
      error: null
    }
  },
  methods: {
    async getBookCount() {
      try {
        const response = await axios.get('https://countbooks-niaadyuddq-uc.a.run.app')
        this.count = response.data.count
        this.error = null
      } catch (error) {
        console.error('Error fetching book count:', error)
        this.error = 'Error fetching book count.'
        this.count = null
      }
    }
```



```
    }
}
</script>
```

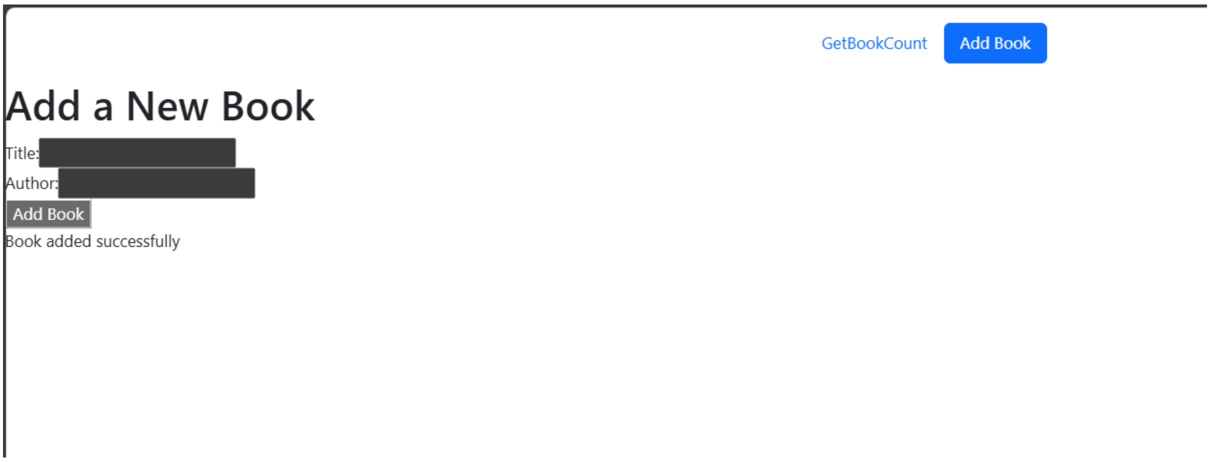
Open the function URL:



```
{
    "count": 7
}
```

Task 9.2

Screenshot set 1:



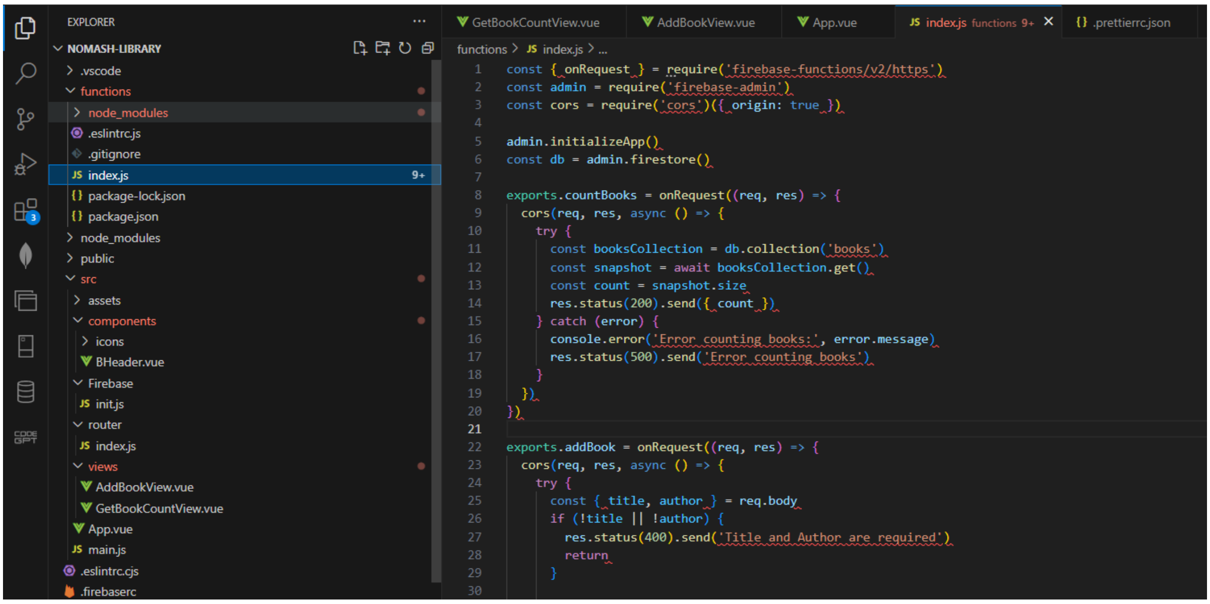
Add a New Book

Title: [redacted]

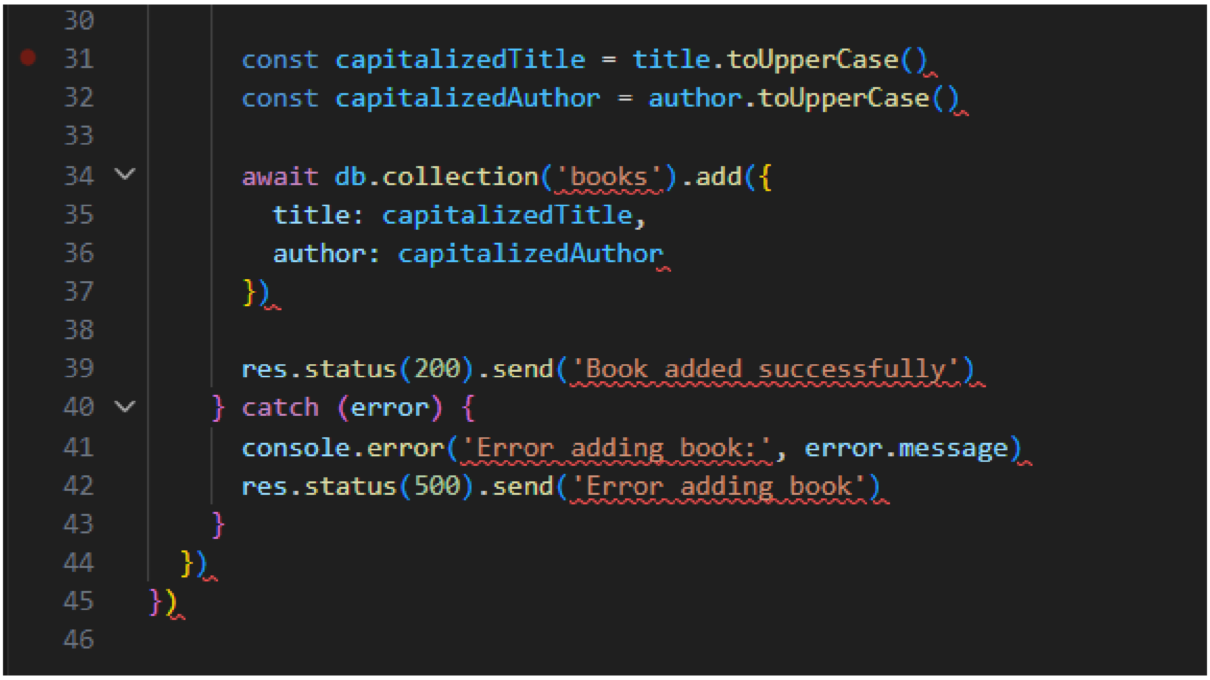
Author: [redacted]

Book added successfully

The **index.js** file in the **functions** folder:

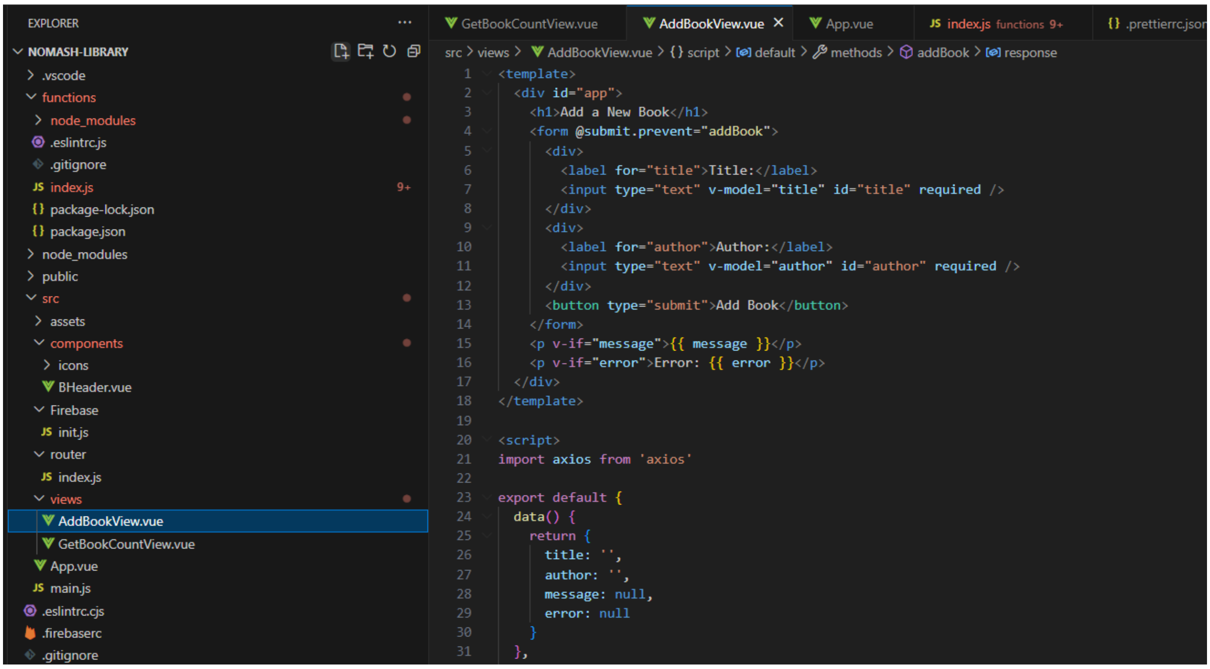


```javascript
const { onRequest } = require('firebase-functions/v2/https')
const admin = require('firebase-admin')
const cors = require('cors')({ origin: true })

admin.initializeApp()
const db = admin.firestore()

exports.countBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = db.collection('books')
      const snapshot = await booksCollection.get()
      const count = snapshot.size
      res.status(200).send({ count })
    } catch (error) {
      console.error('Error counting books:', error.message)
      res.status(500).send('Error counting books')
    }
  })
})

exports.addBook = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const { title, author } = req.body
      if (!title || !author) {
        res.status(400).send('Title and Author are required')
        return
      }

```

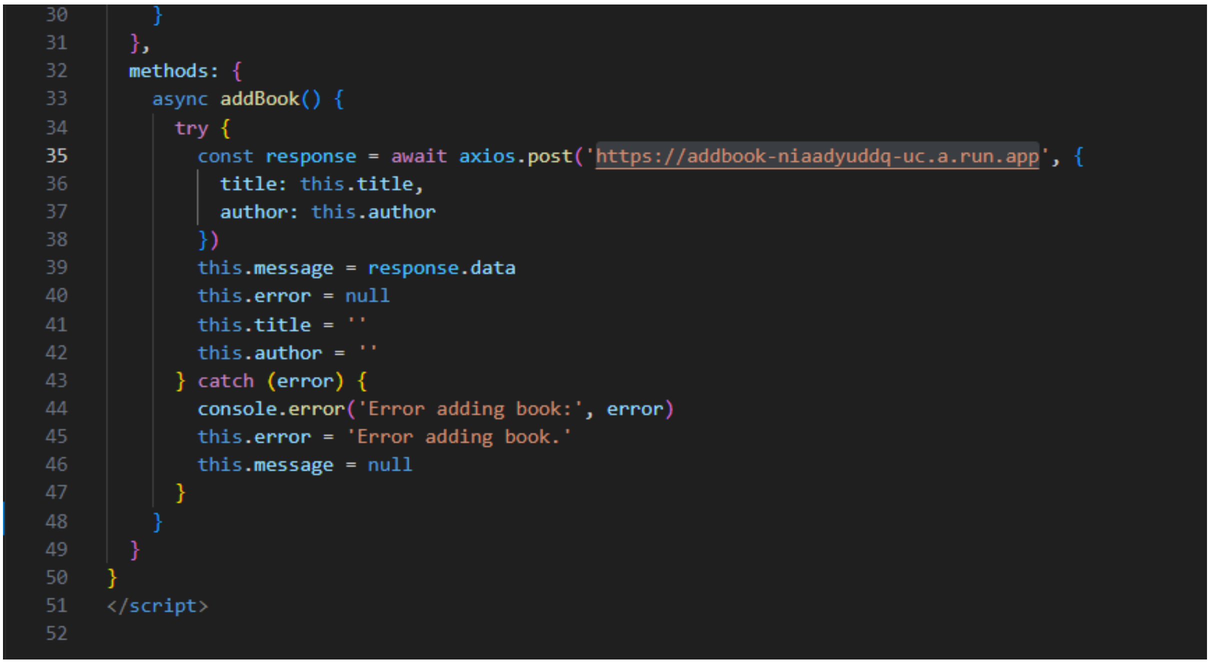


```
   const capitalizedTitle = title.toUpperCase()
   const capitalizedAuthor = author.toUpperCase()

   await db.collection('books').add({
    title: capitalizedTitle,
    author: capitalizedAuthor
   })

   res.status(200).send('Book added successfully')
  } catch (error) {
   console.error('Error adding book:', error.message)
   res.status(500).send('Error adding book')
  }
 })
})
```

AddBookView.vue:

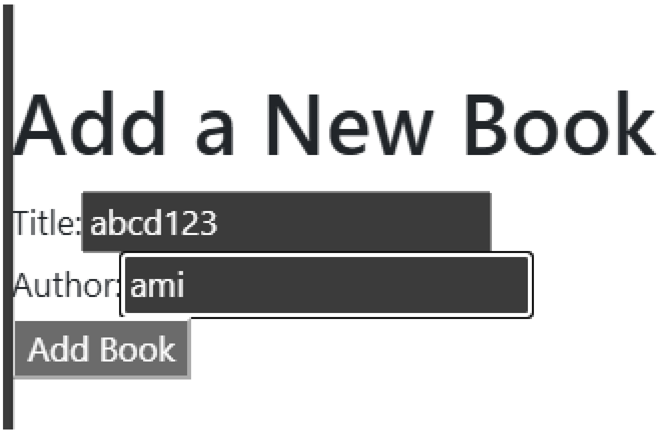


```
<template>
  <div id="app">
    <h1>Add a New Book</h1>
    <form @submit.prevent="addBook">
      <div>
        <label for="title">Title:</label>
        <input type="text" v-model="title" id="title" required />
      </div>
      <div>
        <label for="author">Author:</label>
        <input type="text" v-model="author" id="author" required />
      </div>
      <button type="submit">Add Book</button>
    </form>
    <p v-if="message">{{ message }}</p>
    <p v-if="error">Error: {{ error }}</p>
  </div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      title: '',
      author: '',
      message: null,
      error: null
    }
  },
```



```
    }
  },
  methods: {
    async addBook() {
      try {
        const response = await axios.post('https://addbook-niaadyuddq-uc.a.run.app', {
          title: this.title,
          author: this.author
        })
        this.message = response.data
        this.error = null
        this.title = ''
        this.author = ''
      } catch (error) {
        console.error('Error adding book:', error)
        this.error = 'Error adding book.'
        this.message = null
      }
    }
  }
}
</script>
```

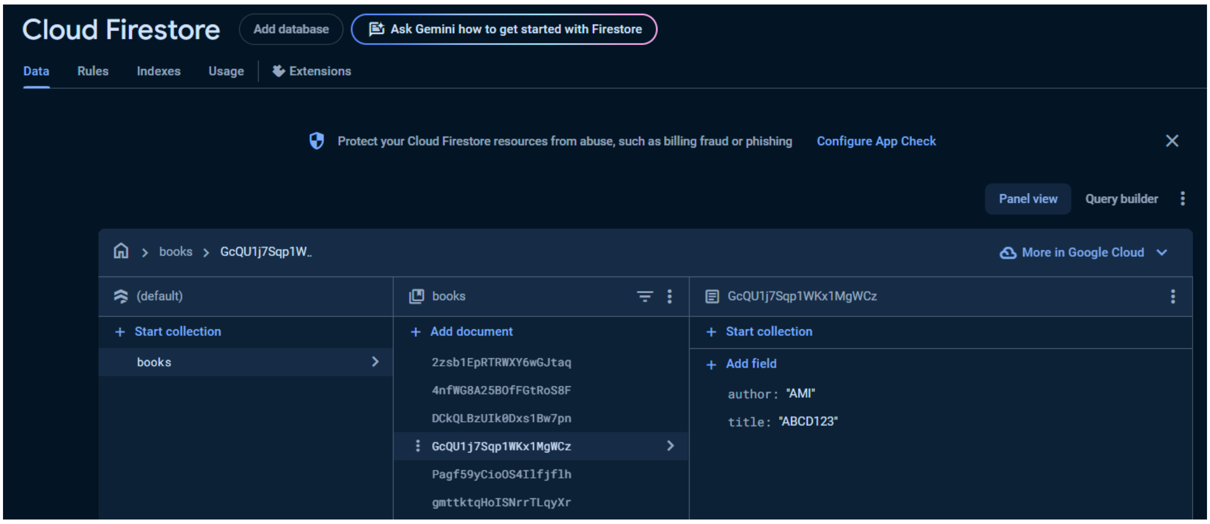
Data stored in the database after uppercase conversion:



Add a New Book

Title: abcd123

Author: ami



Cloud Firestore

Add database | Ask Gemini how to get started with Firestore

Data | Rules | Indexes | Usage | Extensions

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

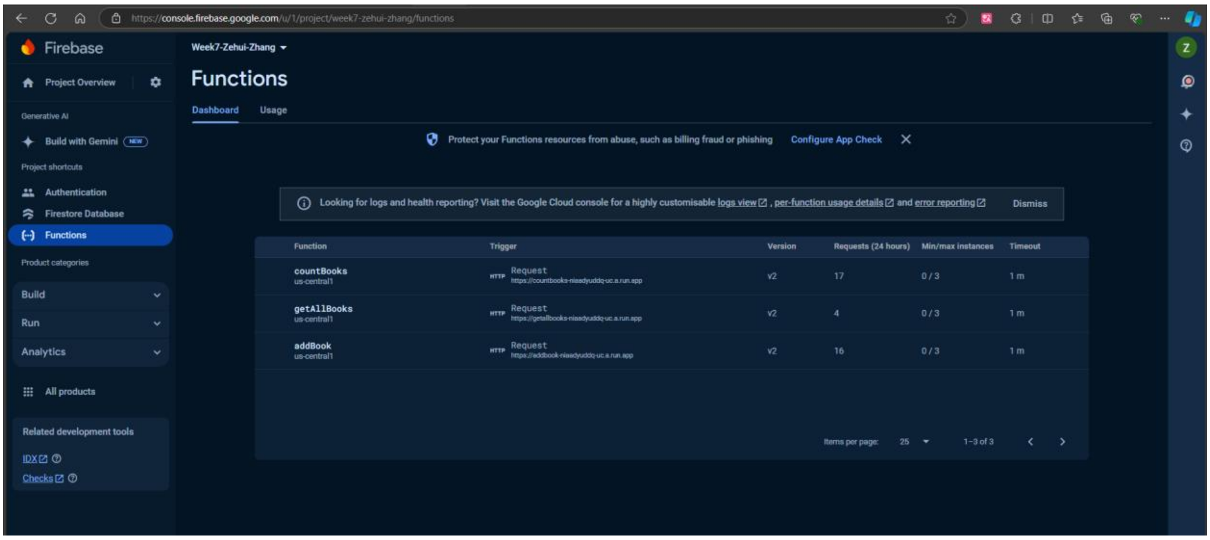
Panel view | Query builder

books > GcQU1j7Sqp1W_

More in Google Cloud

(default)	books	GcQU1j7Sqp1WKx1MgWCz
+ Start collection	+ Add document	+ Start collection
books	2zsb1EpRTRWXY6wGJtaq	+ Add field
	4nfWG8A25BOfFGtRoS8F	author: "AMI"
	DCkQLBzUIk0Dxs1Bw7pn	title: "ABCD123"
	GcQU1j7Sqp1WKx1MgWCz	
	Pagf59yCioOS4Ilfjflh	
	gmttktqHoISNrrTLqyXr	

Deployed function：



https://console.firebase.google.com/u/1/project/week7-zehui-zhang/functions

Firebase

Week7-Zehui-Zhang

Functions

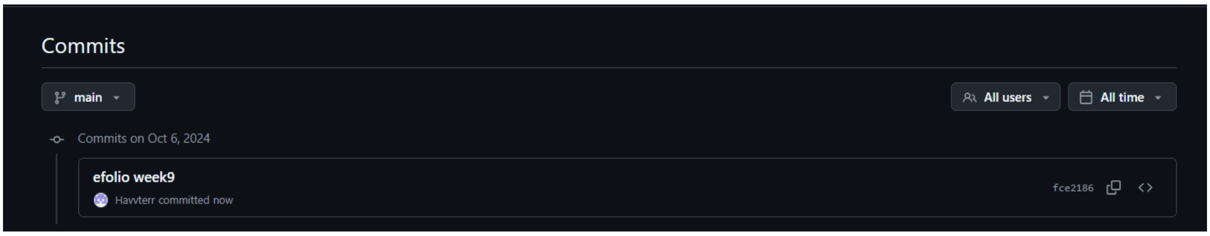
Protect your Functions resources from abuse, such as billing fraud or phishing Configure App Check

Looking for logs and health reporting? Visit the Google Cloud console for a highly customisable logs view , per-function usage details and error reporting Dismiss

Function	Trigger	Version	Requests (24 hours)	Min/max instances	Timeout
countBooks us-central1	HTTP Request https://countbooks-niaadyuddq-uc.a.run.app	v2	17	0 / 3	1 m
getAllBooks us-central1	HTTP Request https://getallbooks-niaadyuddq-uc.a.run.app	v2	4	0 / 3	1 m
addBook us-central1	HTTP Request https://addbook-niaadyuddq-uc.a.run.app	v2	16	0 / 3	1 m

Items per page: 25 1–3 of 3

Commit History:



Commits

main | All users | All time

Commits on Oct 6, 2024

efolio week9

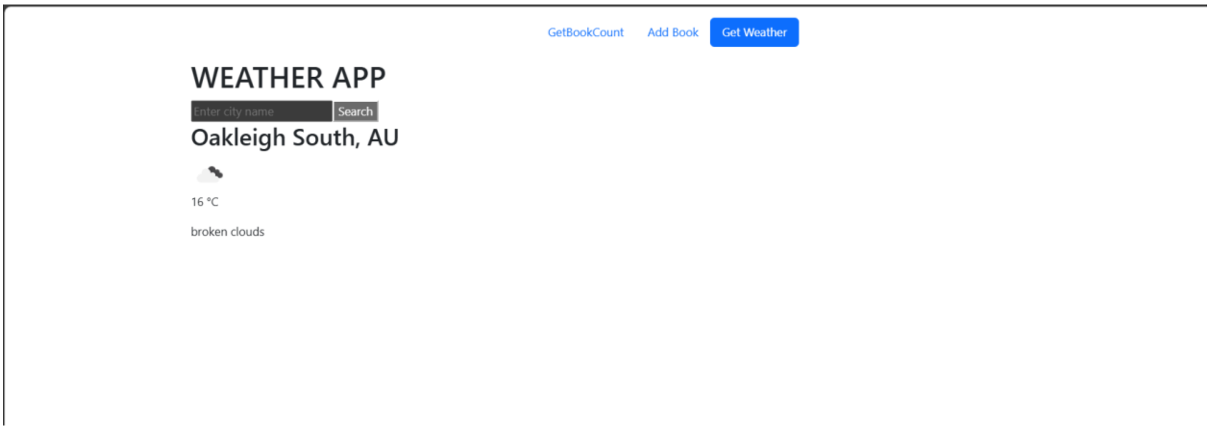
Havterr committed now

fce2186

Efolio week 10

Task 10.1

Screenshot set 1:



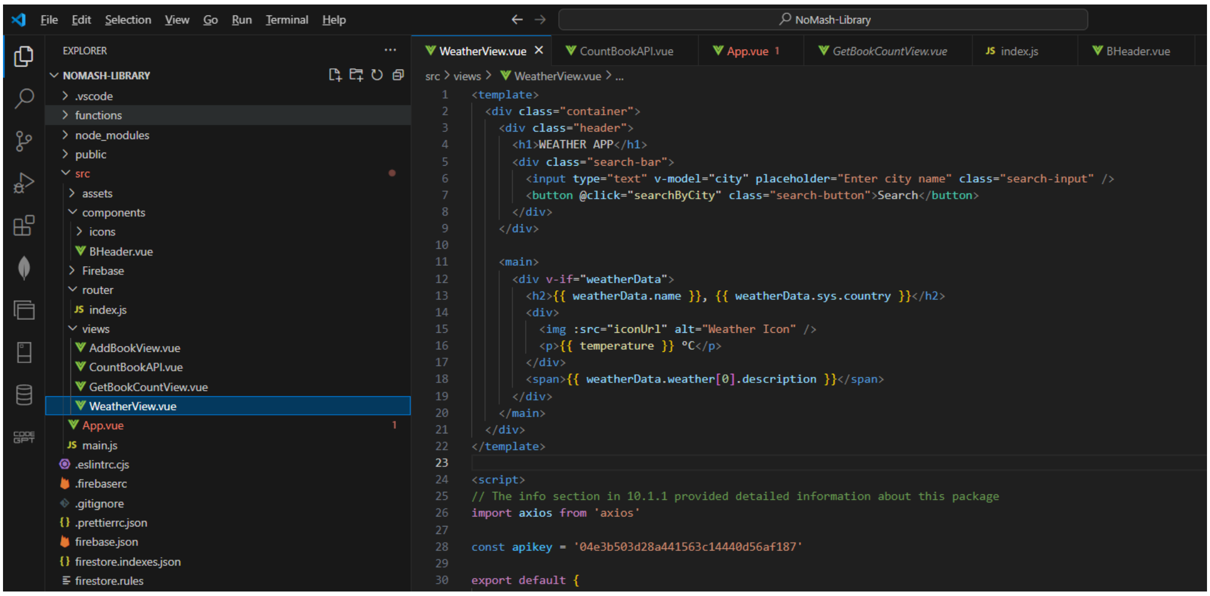
WEATHER APP

Oakleigh South, AU

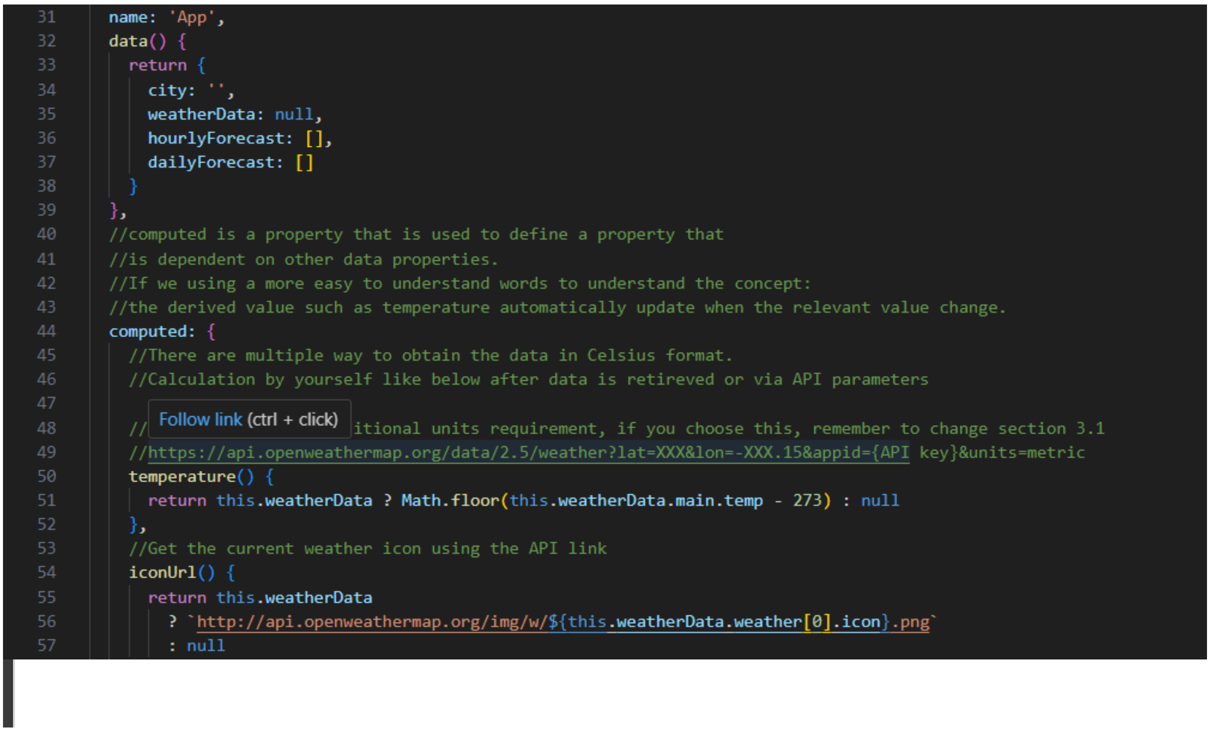
16 °C

broken clouds

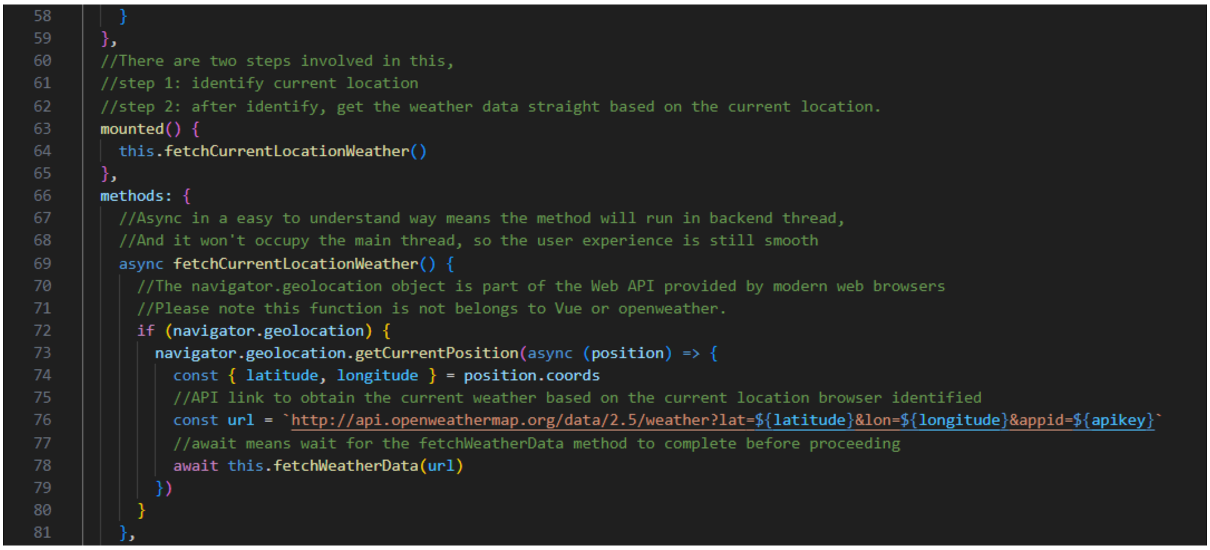
WeatherView.vue:



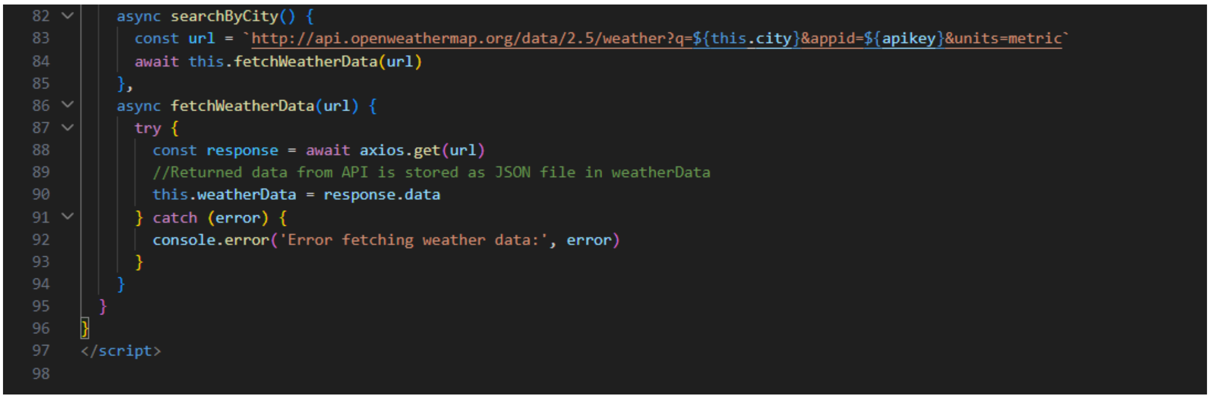
```
<template>
  <div class="container">
    <div class="header">
      <h1>WEATHER APP</h1>
      <div class="search-bar">
        <input type="text" v-model="city" placeholder="Enter city name" class="search-input" />
        <button @click="searchByCity" class="search-button">Search</button>
      </div>
    </div>

    <main>
      <div v-if="weatherData">
        <h2>{{ weatherData.name }}, {{ weatherData.sys.country }}</h2>
        <div>
          <img :src="iconUrl" alt="Weather Icon" />
          <p>{{ temperature }} °C</p>
        </div>
        <span>{{ weatherData.weather[0].description }}</span>
      </div>
    </main>
  </div>
</template>

<script>
// The info section in 10.1.1 provided detailed information about this package
import axios from 'axios'

const apikey = '04e3b503d28a441563c14440d56af187'

export default {
```



```
  name: 'App',
  data() {
    return {
      city: '',
      weatherData: null,
      hourlyForecast: [],
      dailyForecast: []
    }
  },
  //computed is a property that is used to define a property that
  //is dependent on other data properties.
  //If we using a more easy to understand words to understand the concept:
  //the derived value such as temperature automatically update when the relevant value change.
  computed: {
    //There are multiple way to obtain the data in Celsius format.
    //Calculation by yourself like below after data is retireved or via API parameters

    //itional units requirement, if you choose this, remember to change section 3.1
    //https://api.openweathermap.org/data/2.5/weather?lat=XXX&lon=-XXX.15&appid={API key}&units=metric
    temperature() {
      return this.weatherData ? Math.floor(this.weatherData.main.temp - 273) : null
    },
    //Get the current weather icon using the API link
    iconUrl() {
      return this.weatherData
        ? `http://api.openweathermap.org/img/w/${this.weatherData.weather[0].icon}.png`
        : null
```

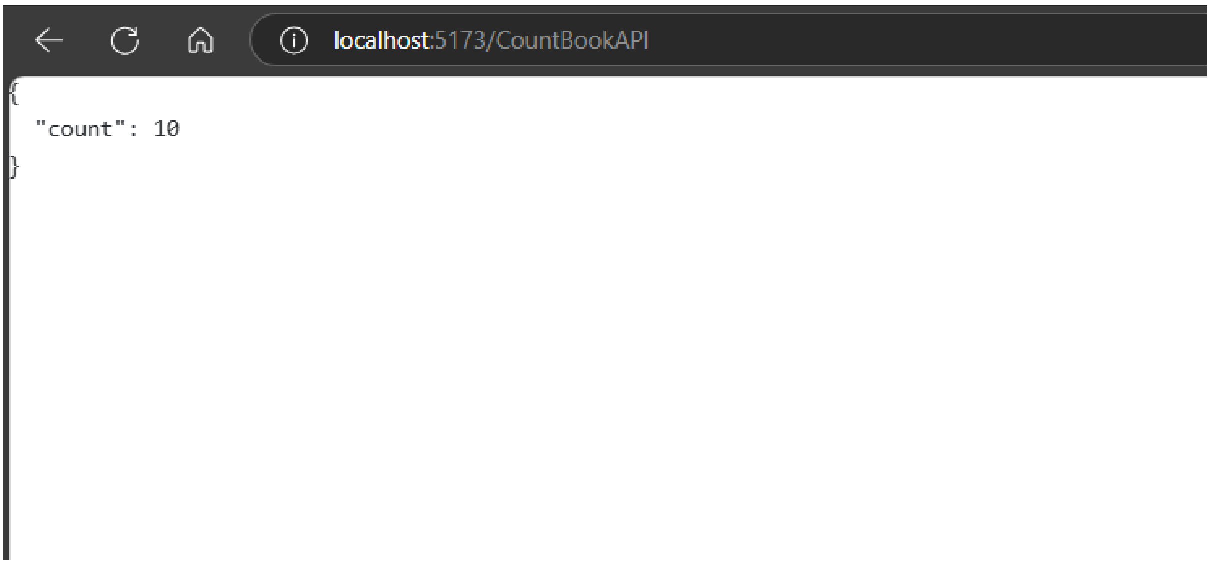


```
    }
  },
  //There are two steps involved in this,
  //step 1: identify current location
  //step 2: after identify, get the weather data straight based on the current location.
  mounted() {
    this.fetchCurrentLocationWeather()
  },
  methods: {
    //Async in a easy to understand way means the method will run in backend thread,
    //And it won't occupy the main thread, so the user experience is still smooth
    async fetchCurrentLocationWeather() {
      //The navigator.geolocation object is part of the Web API provided by modern web browsers
      //Please note this function is not belongs to Vue or openweather.
      if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(async (position) => {
          const { latitude, longitude } = position.coords
          //API link to obtain the current weather based on the current location browser identified
          const url = `http://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apikey}`
          //await means wait for the fetchWeatherData method to complete before proceeding
          await this.fetchWeatherData(url)
        })
      }
    },
```



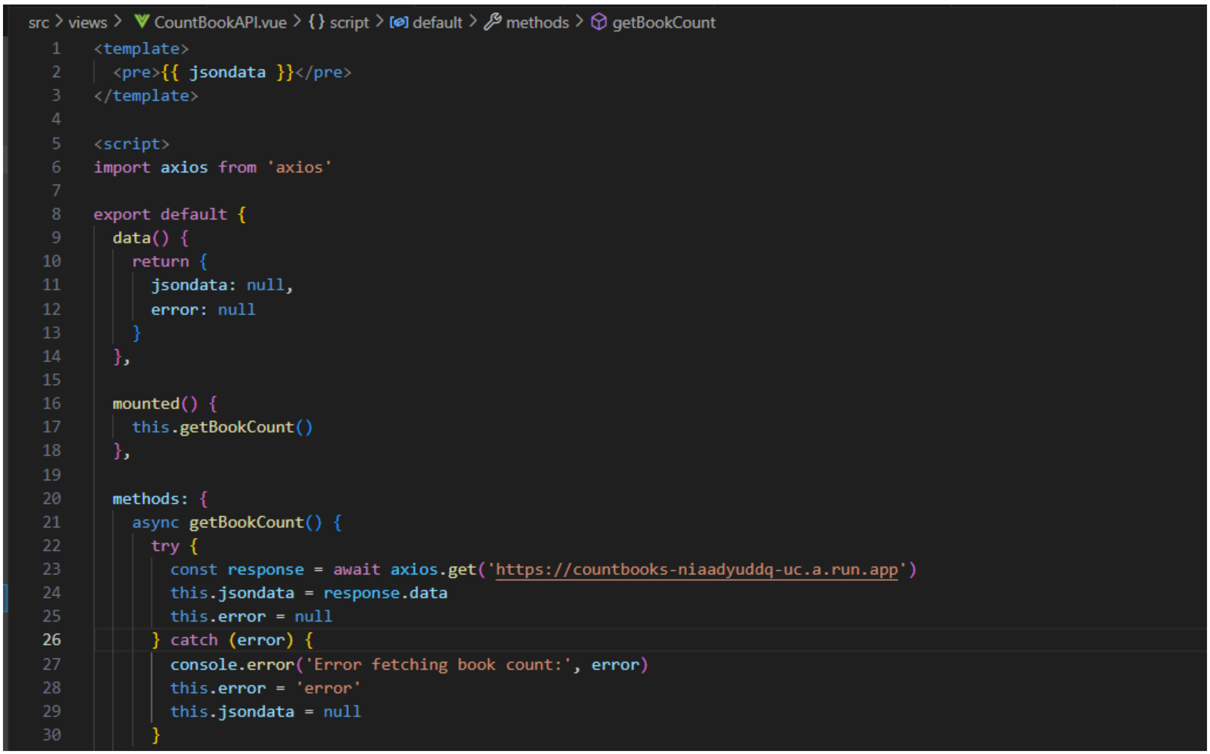
```
    async searchByCity() {
      const url = `http://api.openweathermap.org/data/2.5/weather?q=${this.city}&appid=${apikey}&units=metric`
      await this.fetchWeatherData(url)
    },
    async fetchWeatherData(url) {
      try {
        const response = await axios.get(url)
        //Returned data from API is stored as JSON file in weatherData
        this.weatherData = response.data
      } catch (error) {
        console.error('Error fetching weather data:', error)
      }
    }
  }
}
</script>

```

Screenshot set 2:

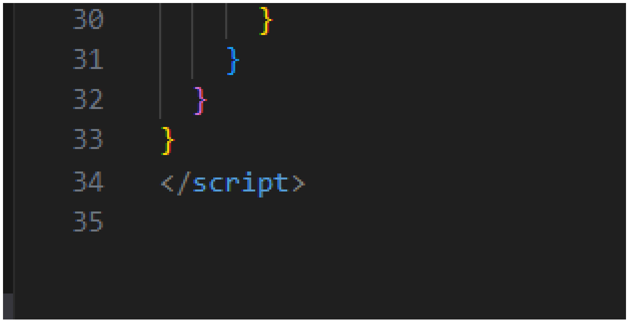


```
{
  "count": 10
}
```

CountBookAPI.vue:

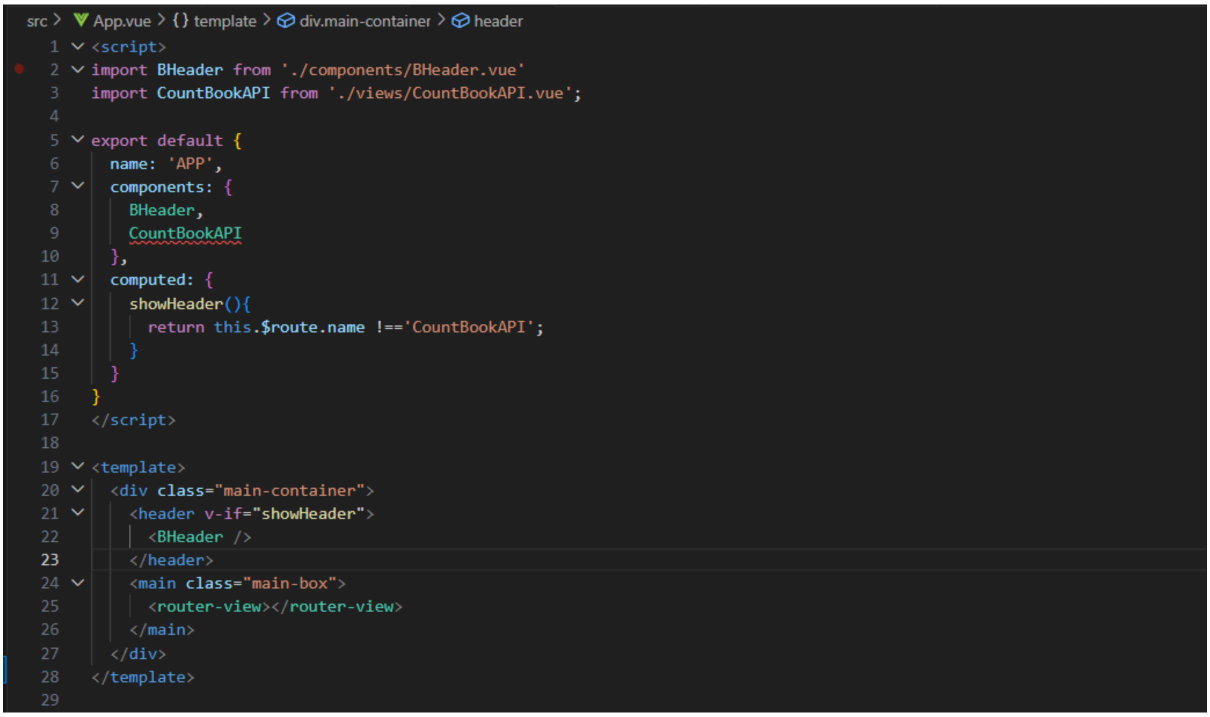

src > views > CountBookAPI.vue > {} script > default > methods > getBookCount

```
<template>
  <pre>{{ jsondata }}</pre>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      jsondata: null,
      error: null
    }
  },

  mounted() {
    this.getBookCount()
  },

  methods: {
    async getBookCount() {
      try {
        const response = await axios.get('https://countbooks-niaadyuddq-uc.a.run.app')
        this.jsondata = response.data
        this.error = null
      } catch (error) {
        console.error('Error fetching book count:', error)
        this.error = 'error'
        this.jsondata = null
      }
```



```
      }
    }
  }
}
</script>

```

App.vue:

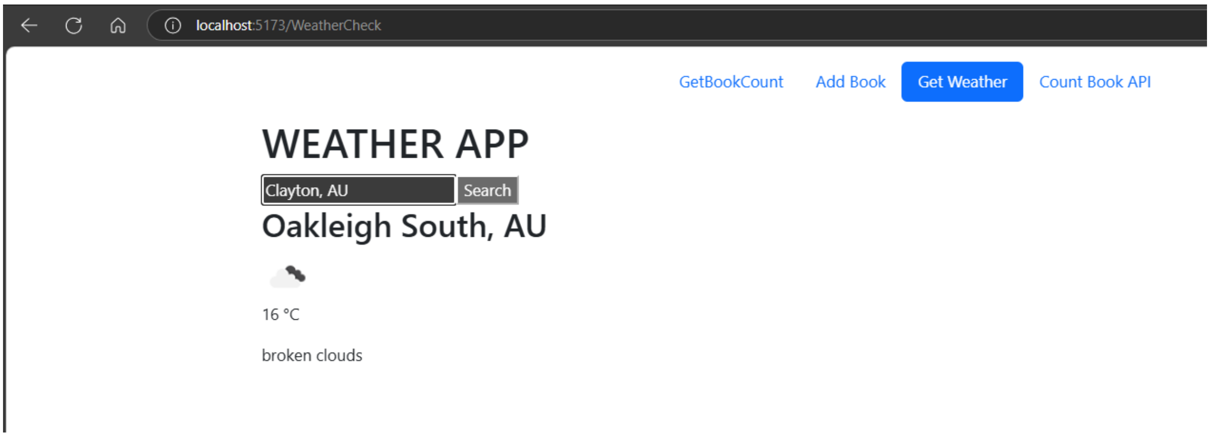


```
<script>
import BHeader from './components/BHeader.vue'
import CountBookAPI from './views/CountBookAPI.vue';

export default {
  name: 'APP',
  components: {
    BHeader,
    CountBookAPI
  },
  computed: {
    showHeader(){
      return this.$route.name !=='CountBookAPI';
    }
  }
}
</script>

<template>
  <div class="main-container">
    <header v-if="showHeader">
      <BHeader />
    </header>
    <main class="main-box">
      <router-view></router-view>
    </main>
  </div>
</template>
```

Task 10.2

Screenshot set 1:

Before performing the search:



WEATHER APP

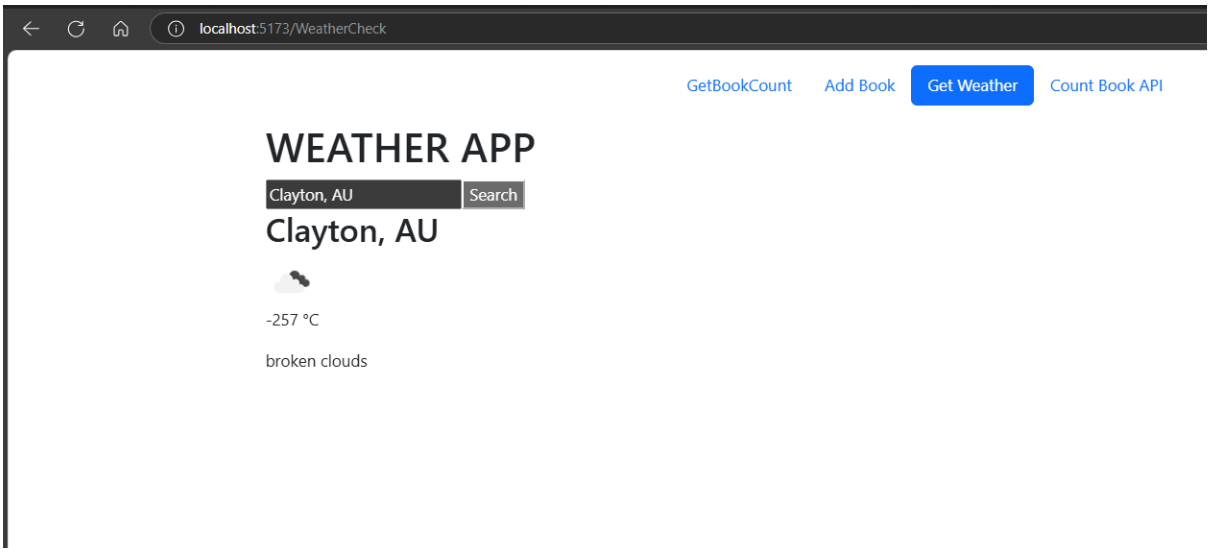
Clayton, AU

Oakleigh South, AU

16 °C

broken clouds

After performing the search:



localhost:5173/WeatherCheck

WEATHER APP

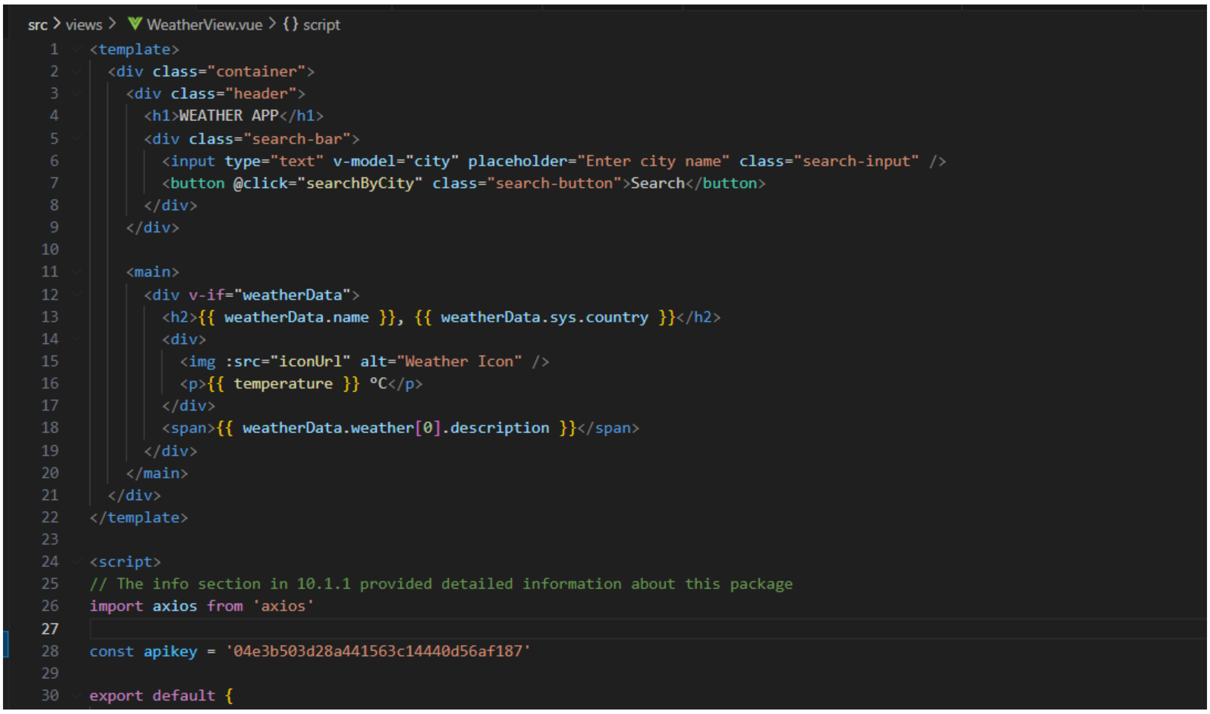
Clayton, AU

Clayton, AU

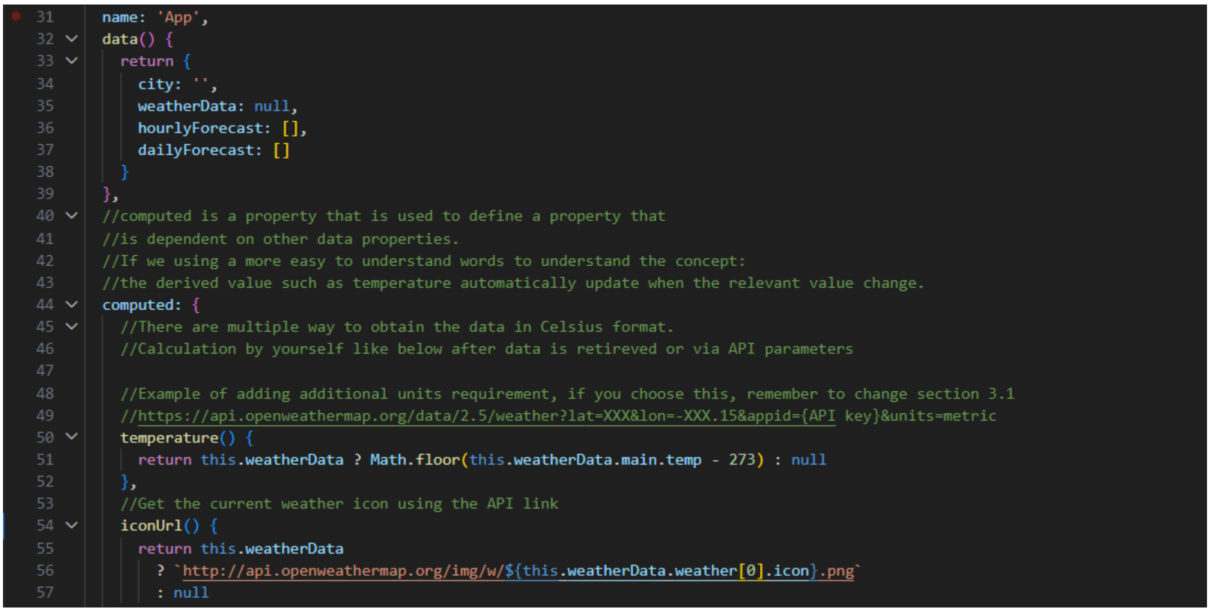
-257 °C

broken clouds

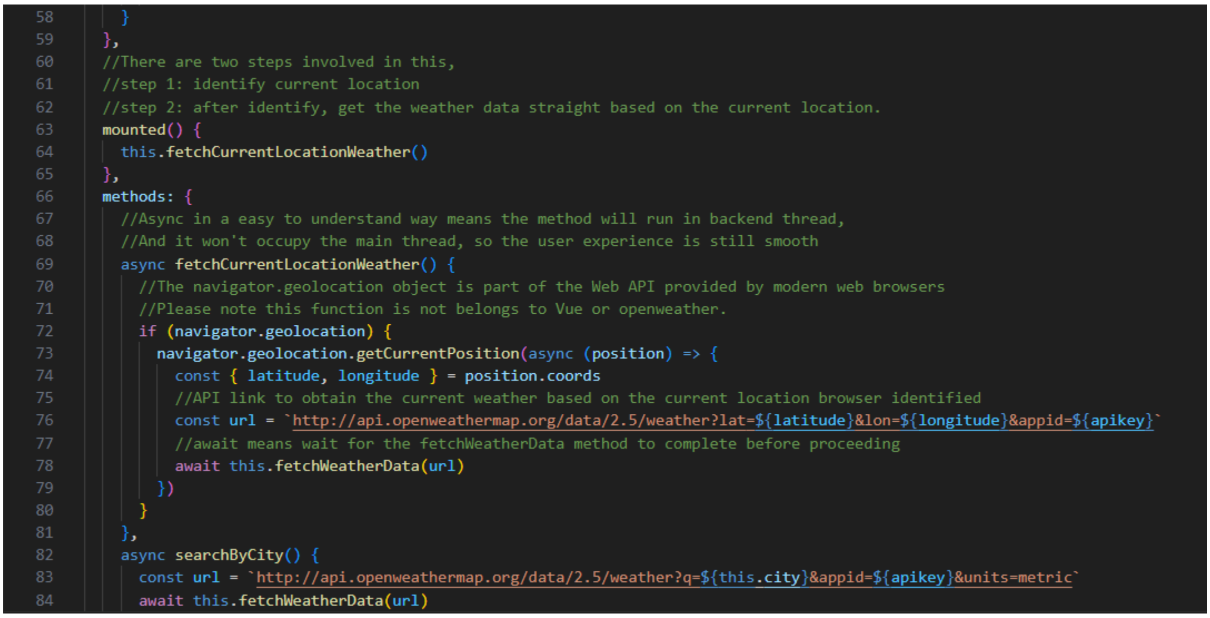
WeatherView.vue:



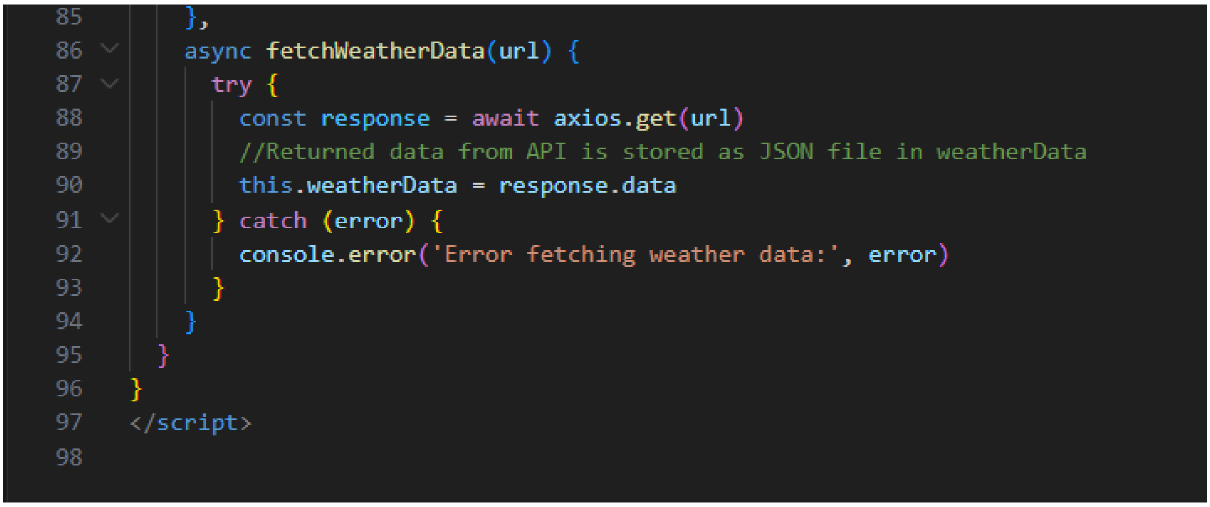
```
<template>
  <div class="container">
    <div class="header">
      <h1>WEATHER APP</h1>
      <div class="search-bar">
        <input type="text" v-model="city" placeholder="Enter city name" class="search-input" />
        <button @click="searchByCity" class="search-button">Search</button>
      </div>
    </div>

    <main>
      <div v-if="weatherData">
        <h2>{{ weatherData.name }}, {{ weatherData.sys.country }}</h2>
        <div>
          <img :src="iconUrl" alt="Weather Icon" />
          <p>{{ temperature }} °C</p>
        </div>
        <span>{{ weatherData.weather[0].description }}</span>
      </div>
    </main>
  </div>
</template>

<script>
// The info section in 10.1.1 provided detailed information about this package
import axios from 'axios'

const apikey = '04e3b503d28a441563c14440d56af187'

export default {
```



```
  name: 'App',
  data() {
    return {
      city: '',
      weatherData: null,
      hourlyForecast: [],
      dailyForecast: []
    }
  },
  //computed is a property that is used to define a property that
  //is dependent on other data properties.
  //If we using a more easy to understand words to understand the concept:
  //the derived value such as temperature automatically update when the relevant value change.
  computed: {
    //There are multiple way to obtain the data in Celsius format.
    //Calculation by yourself like below after data is retrieved or via API parameters

    //Example of adding additional units requirement, if you choose this, remember to change section 3.1
    //https://api.openweathermap.org/data/2.5/weather?lat=XXX&lon=-XXX.15&appid={API key}&units=metric
    temperature() {
      return this.weatherData ? Math.floor(this.weatherData.main.temp - 273) : null
    },
    //Get the current weather icon using the API link
    iconUrl() {
      return this.weatherData
        ? `http://api.openweathermap.org/img/w/${this.weatherData.weather[0].icon}.png`
        : null
```

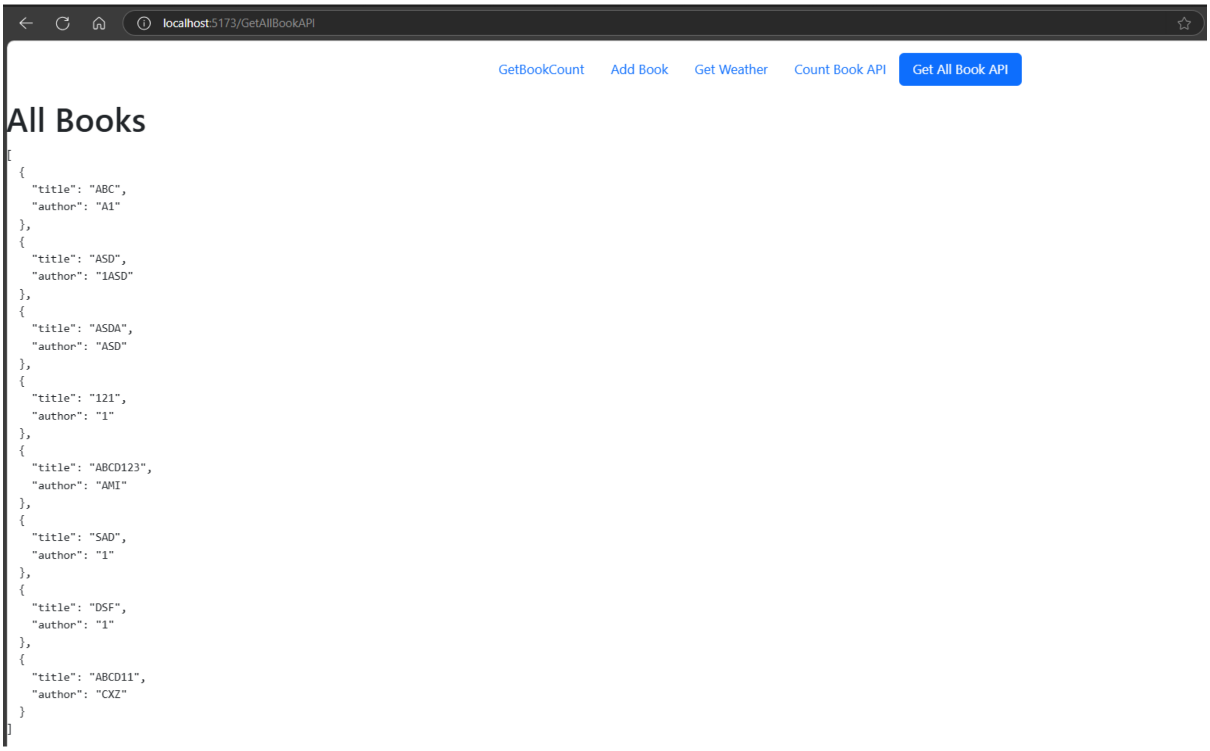


```
    }
  },
  //There are two steps involved in this,
  //step 1: identify current location
  //step 2: after identify, get the weather data straight based on the current location.
  mounted() {
    this.fetchCurrentLocationWeather()
  },
  methods: {
    //Async in a easy to understand way means the method will run in backend thread,
    //And it won't occupy the main thread, so the user experience is still smooth
    async fetchCurrentLocationWeather() {
      //The navigator.geolocation object is part of the Web API provided by modern web browsers
      //Please note this function is not belongs to Vue or openweather.
      if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(async (position) => {
          const { latitude, longitude } = position.coords
          //API link to obtain the current weather based on the current location browser identified
          const url = `http://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apikey}`
          //await means wait for the fetchWeatherData method to complete before proceeding
          await this.fetchWeatherData(url)
        })
      }
    },
    async searchByCity() {
      const url = `http://api.openweathermap.org/data/2.5/weather?q=${this.city}&appid=${apikey}&units=metric`
      await this.fetchWeatherData(url)
```



```
    },
    async fetchWeatherData(url) {
      try {
        const response = await axios.get(url)
        //Returned data from API is stored as JSON file in weatherData
        this.weatherData = response.data
      } catch (error) {
        console.error('Error fetching weather data:', error)
      }
    }
  }
}
</script>

```

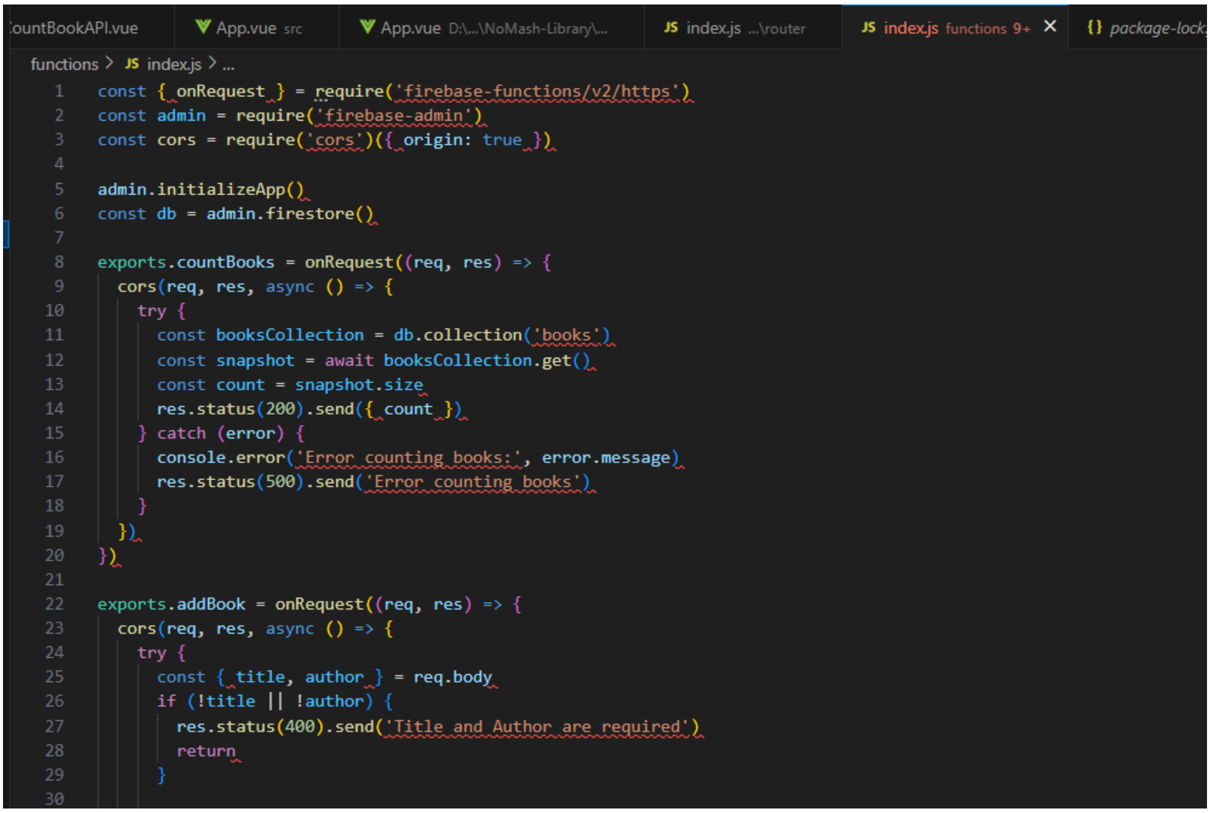
Screenshot set 2:



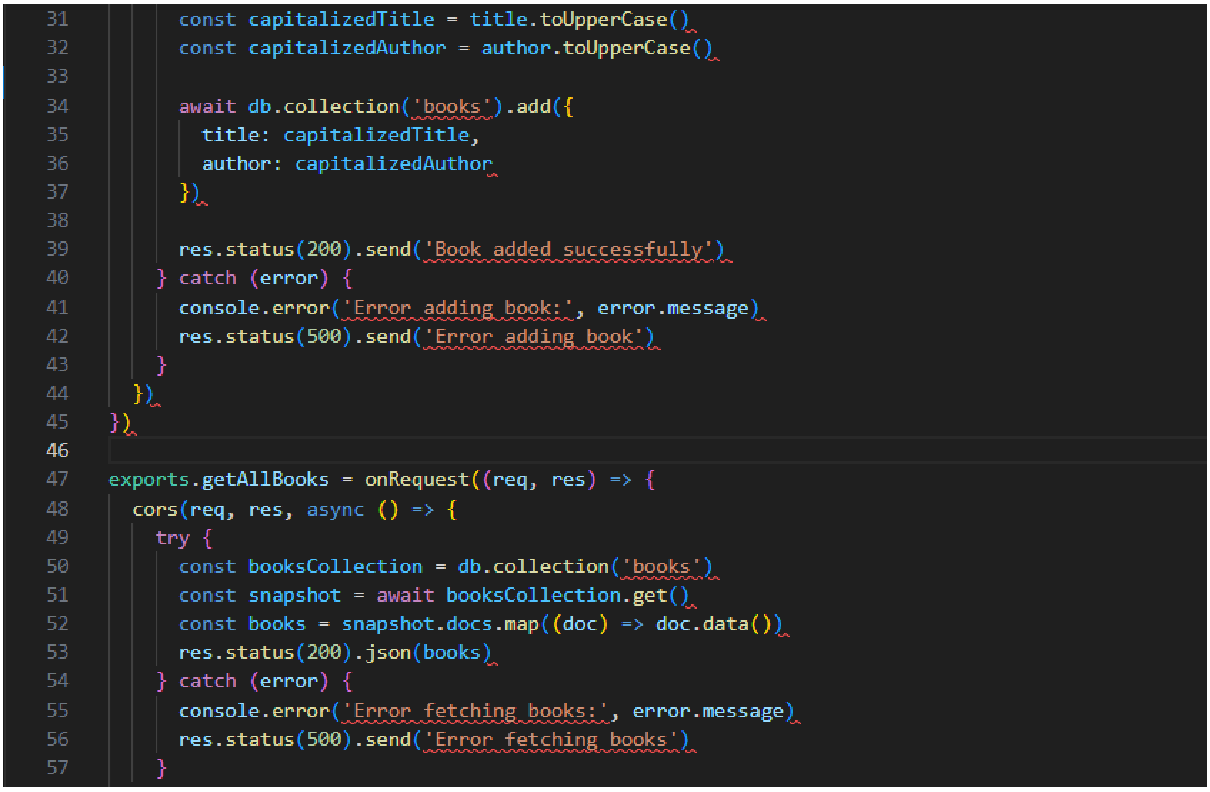
All Books

```
[
  {
    "title": "ABC",
    "author": "A1"
  },
  {
    "title": "ASD",
    "author": "1ASD"
  },
  {
    "title": "ASDA",
    "author": "ASD"
  },
  {
    "title": "121",
    "author": "1"
  },
  {
    "title": "ABCD123",
    "author": "AMI"
  },
  {
    "title": "SAD",
    "author": "1"
  },
  {
    "title": "DSF",
    "author": "1"
  },
  {
    "title": "ABCD11",
    "author": "CXZ"
  }
]
```

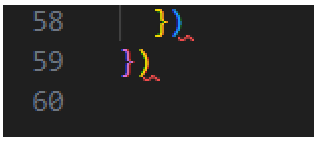
The **index.js** file in the **functions** folder:



```javascript
const { onRequest } = require('firebase-functions/v2/https')
const admin = require('firebase-admin')
const cors = require('cors')({ origin: true })

admin.initializeApp()
const db = admin.firestore()

exports.countBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = db.collection('books')
      const snapshot = await booksCollection.get()
      const count = snapshot.size
      res.status(200).send({ count })
    } catch (error) {
      console.error('Error counting books:', error.message)
      res.status(500).send('Error counting books')
    }
  })
})

exports.addBook = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const { title, author } = req.body
      if (!title || !author) {
        res.status(400).send('Title and Author are required')
        return
      }

```

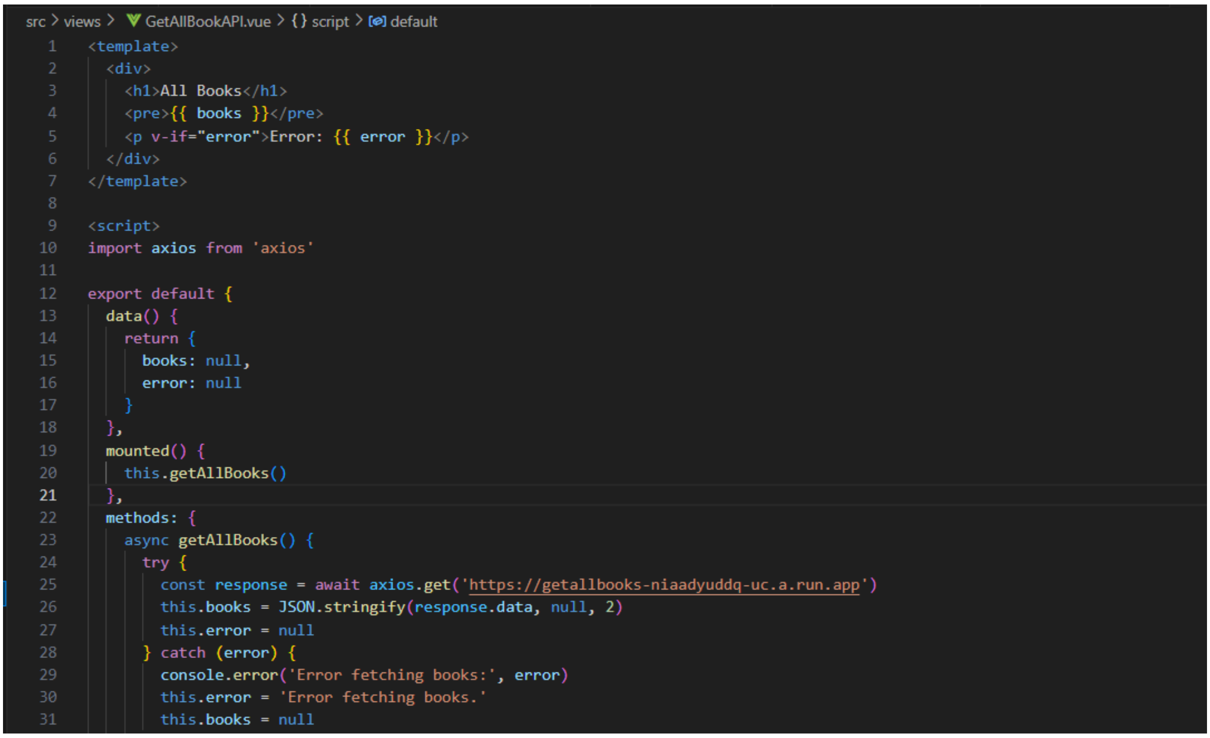


```javascript
      const capitalizedTitle = title.toUpperCase()
      const capitalizedAuthor = author.toUpperCase()

      await db.collection('books').add({
        title: capitalizedTitle,
        author: capitalizedAuthor
      })

      res.status(200).send('Book added successfully')
    } catch (error) {
      console.error('Error adding book:', error.message)
      res.status(500).send('Error adding book')
    }
  })
})

exports.getAllBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = db.collection('books')
      const snapshot = await booksCollection.get()
      const books = snapshot.docs.map((doc) => doc.data())
      res.status(200).json(books)
    } catch (error) {
      console.error('Error fetching books:', error.message)
      res.status(500).send('Error fetching books')
    }
```



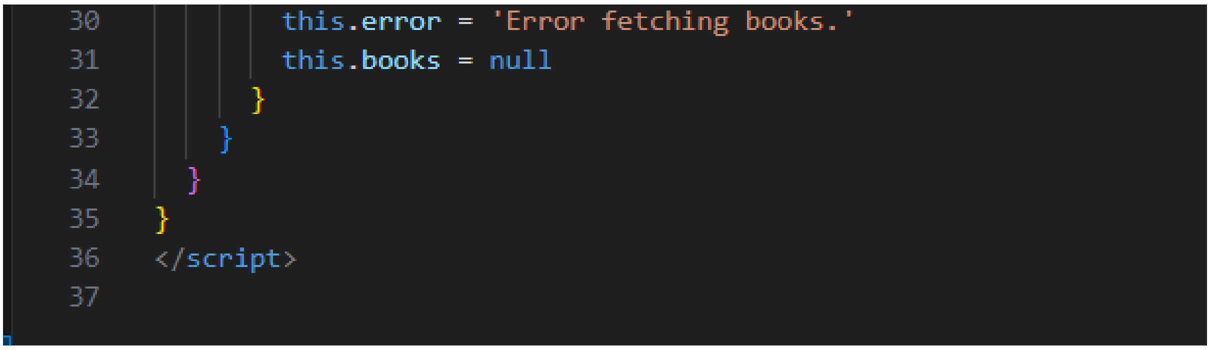
```javascript
  })
})

```

GetAllBookAPI.vue



src > views > GetAllBookAPI.vue > {} script > default

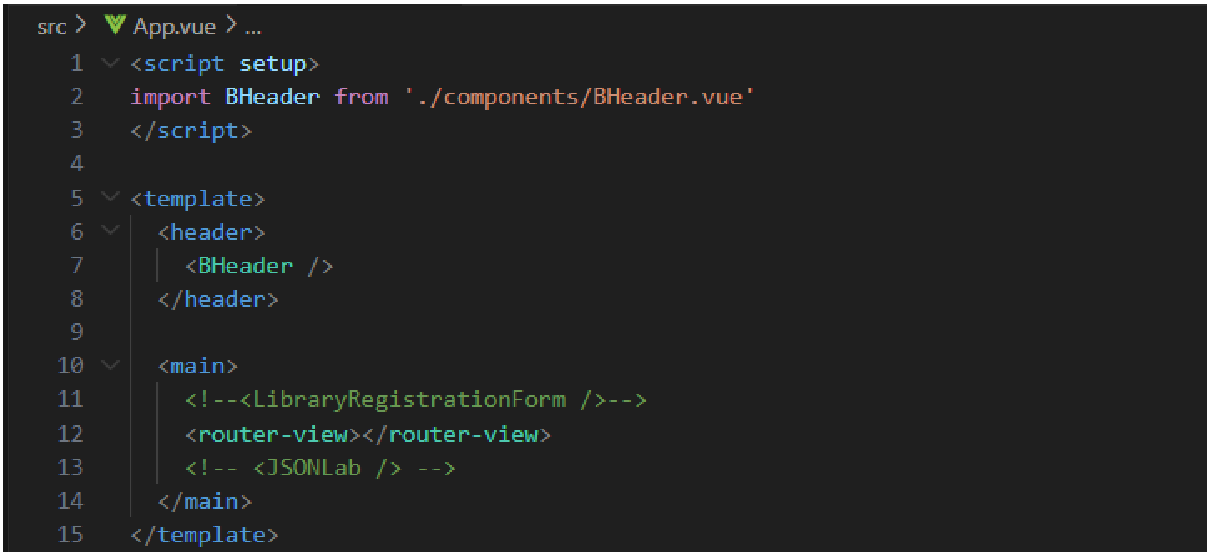
```
<template>
  <div>
    <h1>All Books</h1>
    <pre>{{ books }}</pre>
    <p v-if="error">Error: {{ error }}</p>
  </div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      books: null,
      error: null
    }
  },
  mounted() {
    this.getAllBooks()
  },
  methods: {
    async getAllBooks() {
      try {
        const response = await axios.get('https://getallbooks-niaadyuddq-uc.a.run.app')
        this.books = JSON.stringify(response.data, null, 2)
        this.error = null
      } catch (error) {
        console.error('Error fetching books:', error)
        this.error = 'Error fetching books.'
        this.books = null
```



```
        this.error = 'Error fetching books.'
        this.books = null
```

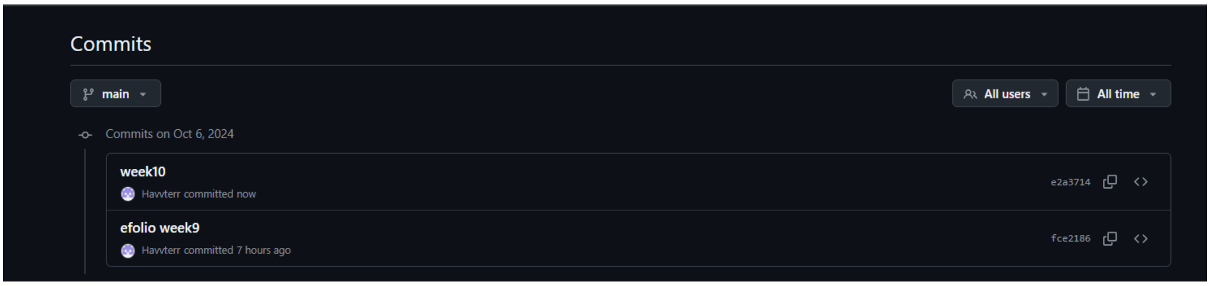
```
      }
    }
  }
}
</script>

```

App.vue:



```
<script setup>
import BHeader from './components/BHeader.vue'
</script>

<template>
  <header>
    <BHeader />
  </header>

  <main>
    <!--<LibraryRegistrationForm />-->
    <router-view></router-view>
    <!-- <JSONLab /> -->
  </main>
</template>
```

Commit History:



Commits on Oct 6, 2024

week10
Havterr committed now
e2a3714

efolio week9
Havterr committed 7 hours ago
fce2186