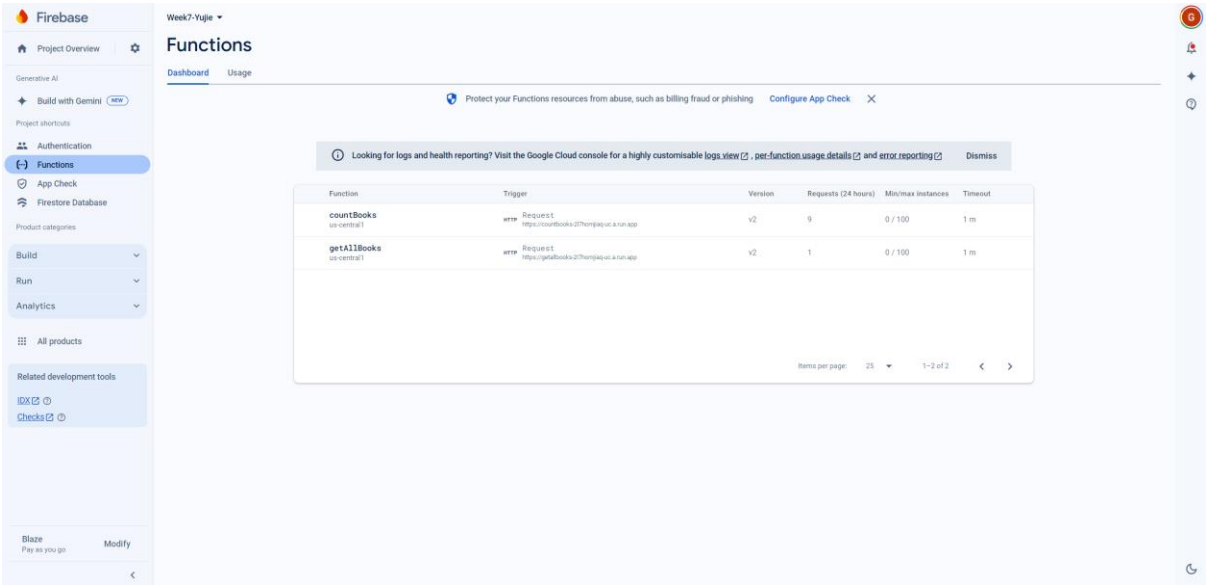
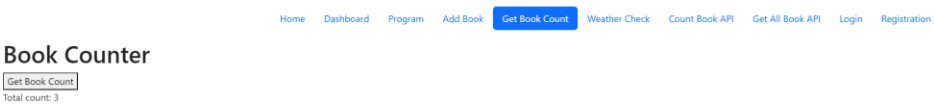
Efolio Task 9.1





Book Counter

Get Book Count

Total count: 3

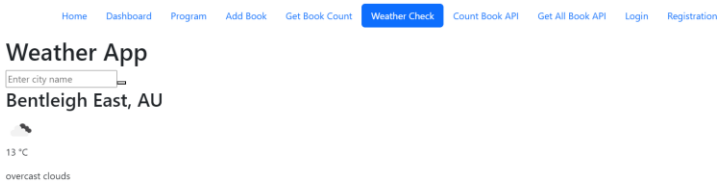
AddBookView.vue | BookList.vue | package.json | WeatherView.vue | CountBookAPI.vue | GetAllBookAPI.vue | LibraryRegistrationForm.vue

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

```vue
<template>
    <h1>Book Counter</h1>
    <button @click="getBookCountAPI">Get Book Count</button>
    <p v-if="jsondata !== null">{{ jsondata }}</p>
    <p v-if="error">{{ error }}</p>
</template>

<script>
import axios from 'axios';

export default{
    data(){
        return{
            jsondata: null,
            error: null
        }
    },
    methods: {
        async getBookCountAPI(){
            try {
                const response = await axios.get('https://countbooks-2l7homjiaq-uc.a.run.app')
                this.jsondata = response.data
                this.error = null
            }
            catch (error){
                console.error('Error fetching book count:', error)
                this.error = error
                this.count = null
            }
        }
    }
}
</script>
```

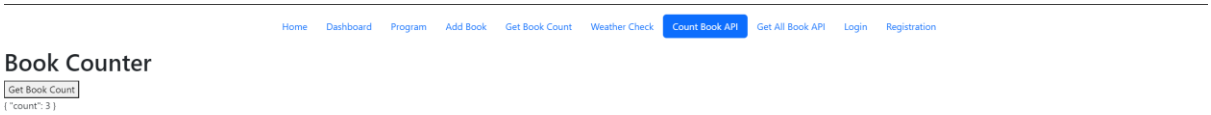
```javascript
const {onRequest} = require("firebase-functions/v2/https");
// const logger = require("firebase-functions/logger");

const admin = require("firebase-admin");
const cors = require("cors")({origin: true});

admin.initializeApp();

exports.countBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = admin.firestore().collection("books");
      const snapshot = await booksCollection.get();
      const count = snapshot.size;

      res.status(200).send({count});
    } catch (error) {
      console.error("Error counting books: ", error);
      res.status(500).send("Error counting books: " + error.message);
    }
  });
});
```

Efolio Task 10.1




```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"></button>

                    <!--The <main> tag in HTML is used to specify the main content of a document
                More info about main, check https://www.w3schools.com/tags/tag_main.asp-->
                <main>
                    <!--If there are no data returned, then skip rendering the information-->
                    <div v-if="weatherData">
                    <!--Display the weather data attribute returned from API
                    Example of API data: https://openweathermap.org/current-->
                    <h2>
                        {{ weatherData.name }}, {{ weatherData.sys.country }}
                    </h2>
                    <div>
                        <!--The image source of of the weather icon will be coming from a function called iconUrl
                            which will be shared in script later-->
                        <img :src="iconUrl" alt="Weather Icon" />
                        <p>{{ temperature }} °C</p>
                    </div>
                    <!-- weather[0] means the current weather, the way we need to obtain data depends on how
                    the API JSON data looks-->
                    <span>{{ weatherData.weather[0].description }}</span>
                    </div>
                </main>
            </div>
        </div>
    </div>
</template>
```

```vue
<script>
  // The info section in 10.1.1 provided detailed information about this package
  import axios from "axios";

  const apikey = "5c350c3e1bee12f10bf972c0676e503f";

  export default {
    name: "App",
    data() {
      return {
        city: "",
        weatherData: null,
        hourlyForecast: [],
        dailyForecast: [],
      };
    },
    computed: {
      temperature() {
        return this.weatherData
          ? Math.floor(this.weatherData.main.temp - 273)
          : null;
      },
      iconUrl() {
        return this.weatherData
          ? `http://api.openweathermap.org/img/w/${this.weatherData.weather[0].icon}.png`
          : null;
      },
    },
    mounted() {
      this.fetchCurrentLocationWeather();
    },
    methods: {
      async fetchCurrentLocationWeather() {
        if (navigator.geolocation) {
          navigator.geolocation.getCurrentPosition(async (position) => {
            const { latitude, longitude } = position.coords;
            const url = `http://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apikey}`;
            await this.fetchWeatherData(url);
          });
        }
      },
      async searchByCity() {
        if (this.city) {
          const url = `http://api.openweathermap.org/data/2.5/weather?q=${this.city}&appid=${apikey}`;
          await this.fetchWeatherData(url);
        } else {
          console.error("Please enter a city name");
        }
      },
      async fetchWeatherData(url) {
        try {
          const response = await axios.get(url);
          this.weatherData = response.data;
        } catch (error) {
          console.error("Error fetching weather data:", error);
        }
      }
    }
  }
      </script>
```



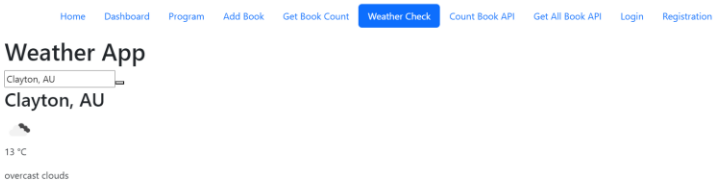
Book Counter

Get Book Count

{ "count": 3 }

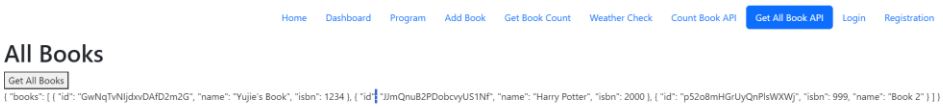
```vue
<template>
    <h1>Book Counter</h1>
    <button @click="getBookCountAPI">Get Book Count</button>
    <p v-if="jsondata !== null">{{ jsondata }}</p>
    <p v-if="error">{{ error }}</p>
</template>

<script>
import axios from 'axios';

export default{
    data(){
        return{
            jsondata: null,
            error: null
        }
    },
    methods: {
        async getBookCountAPI(){
            try {
                const response = await axios.get('https://countbooks-217homjiaq-uc.a.run.app')
                this.jsondata = response.data
                this.error = null
            }
            catch (error){
                console.error('Error fetching book count:', error)
                this.error = error
                this.count = null
            }
        }
    }
}
</script>
```

Efolio Task 10.2




```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"></button>

                    <!--The <main> tag in HTML is used to specify the main content of a document
                More info about main, check https://www.w3schools.com/tags/tag_main.asp-->
                <main>
                    <!--If there are no data returned, then skip rendering the information-->
                    <div v-if="weatherData">
                    <!--Display the weather data attribute returned from API
                    Example of API data: https://openweathermap.org/current-->
                    <h2>
                        {{ weatherData.name }}, {{ weatherData.sys.country }}
                    </h2>
                    <div>
                        <!--The image source of of the weather icon will be coming from a function called iconUrl
                            which will be shared in script later-->
                        <img :src="iconUrl" alt="Weather Icon" />
                        <p>{{ temperature }} °C</p>
                    </div>
                    <!-- weather[0] means the current weather, the way we need to obtain data depends on how
                    the API JSON data looks-->
                    <span>{{ weatherData.weather[0].description }}</span>
                    </div>
                </main>
            </div>
        </div>
    </div>
</template>
```

```vue
<script>
  // The info section in 10.1.1 provided detailed information about this package
  import axios from "axios";

  const apikey = "5c350c3e1bee12f10bf972c0676e503f";

  export default {
    name: "App",
    data() {
      return {
        city: "",
        weatherData: null,
        hourlyForecast: [],
        dailyForecast: [],
      };
    },
    computed: {
      temperature() {
        return this.weatherData
          ? Math.floor(this.weatherData.main.temp - 273)
          : null;
      },
      iconUrl() {
        return this.weatherData
          ? `http://api.openweathermap.org/img/w/${this.weatherData.weather[0].icon}.png`
          : null;
      },
    },
    mounted() {
      this.fetchCurrentLocationWeather();
    },
    methods: {
      async fetchCurrentLocationWeather() {
        if (navigator.geolocation) {
          navigator.geolocation.getCurrentPosition(async (position) => {
            const { latitude, longitude } = position.coords;
            const url = `http://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apikey}`;
            await this.fetchWeatherData(url);
          });
        }
      },
      async searchByCity() {
        if (this.city) {
          const url = `http://api.openweathermap.org/data/2.5/weather?q=${this.city}&appid=${apikey}`;
          await this.fetchWeatherData(url);
        } else {
          console.error("Please enter a city name");
        }
      },
      async fetchWeatherData(url) {
        try {
          const response = await axios.get(url);
          this.weatherData = response.data;
        } catch (error) {
          console.error("Error fetching weather data:", error);
        }
      }
    }
  }
      </script>
```




```vue
<template>
    <h1>All Books</h1>
    <button @click="getAllBookAPI">Get All Books</button>
    <p v-if="jsondata !== null">{{ jsondata }}</p>
    <p v-if="error">{{ error }}</p>
</template>

<script>
import axios from 'axios';

export default{
    data(){
        return{
            jsondata: null,
            error: null
        }
    },
    methods: {
        async getAllBookAPI(){
            try {
                const response = await axios.get('https://getallbooks-2l7homjiaq-uc.a.run.app')
                this.jsondata = response.data
                this.error = null
            }
            catch (error){
                console.error('Error fetching book count:', error)
                this.error = error
                this.count = null
            }
        }
    }
}
</script>
```

```
const {onRequest} = require("firebase-functions/v2/https");
// const logger = require("firebase-functions/logger");

const admin = require("firebase-admin");
const cors = require("cors")({origin: true});

admin.initializeApp();

exports.countBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = admin.firestore().collection("books");
      const snapshot = await booksCollection.get();
      const count = snapshot.size;

      res.status(200).send({count});
    } catch (error) {
      console.error("Error counting books: ", error);
      res.status(500).send("Error counting books: " + error.message);
    }
  });
});

exports.getAllBooks = onRequest((req, res) => {
  cors(req, res, async () => {
    try {
      const booksCollection = admin.firestore().collection("books");
      const snapshot = await booksCollection.get();
      const books = [];
      snapshot.forEach((doc) => {
        books.push({id: doc.id, ...doc.data()});
      });

      // Send the array of books as a response
      res.status(200).send({books});
    } catch (error) {
      console.error("Error fetching books: ", error);
      res.status(500).send("Error fetching books: " + error.message);
    }
  });
});
```