

Projecte JAVASCRIPT



Professor: Jordi Vega

Assignatura: M7 Desenvolupament d'interfícies

Nom: Harvey

Cognoms: John Glover

INDEX:

PÀGINA PRINCIPAL	3
PÀGINA	3
CODI	3
CALCULADORA	6
PÀGINA	6
CODI	6
TESTS	8
API FETCH	9
PÀGINA	9
CODI	9
TESTS	11
API AXIOS	13
PÀGINA	13
CODI	13
TESTS	16
EXTRA	17
HEADER/NAV	17
FOOTER	20
RESPONSIVE	20
SERVIDOR	22
NETIFY	22

PÀGINA PRINCIPAL

PÀGINA

A la pàgina principal veurem el nav, footer i blocs principals que mostren info.



CODI

Aquí mostro el primer bloc que conté el títol i descripció.

```
<!-- Bloc Aplicació SPA amb Vue -->
<div class="lg:pr-4">
  <!-- Primer Bloc Vue -->
  <div class="relative flex flex-col items-center overflow-hidden rounded-3xl bg-[#4a4a4a] px-6 pb-6 pt-4 text-center shadow-2xl sm:px-10 lg:max-w-
    <h1 class="text-4xl font-serif font-bold text-white tracking-tight sm:text-4xl z-10">
      Aplicació SPA amb Vue
    </h1>
    <p class="mt-4 text-lg text-white z-10">
      Aquest projecte és una aplicació web tipus SPA creada amb Vue, que inclou diverses pàgines i funcionalitats per mostrar la capacitat de Vue.
    </p>
    </div>
  </div>
```

Aquí mostro el segon bloc que conté funcions basiques.


```
<!-- Segon Bloc Vue amb funcions destacades -->
<div class="relative flex flex-col items-center overflow-hidden rounded-3xl bg-[#4a4a4a] px-6 pb-6 pt-4 text-center shadow-2xl">
  <h2 class="text-4xl font-serif font-bold text-white tracking-tight sm:text-4xl z-10">
    Funcions Clau
  </h2>
  <p class="mt-4 text-lg text-white z-10">
    Descobreix les funcionalitats principals d'aquesta aplicació SPA amb Vue.
  </p>

  <div class="mt-6 grid grid-cols-2 gap-8 text-white z-10 w-full">
    <div class="flex flex-col items-center">
      
      <span class="mt-2">Rendiment Ràpid</span>
    </div>
    <div class="flex flex-col items-center">
      
      <span class="mt-2">Gestió de Dades</span>
    </div>
    <div class="flex flex-col items-center">
      
      <span class="mt-2">Suport en Temps Real</span>
    </div>
    <div class="flex flex-col items-center">
      
      <span class="mt-2">Responsive</span>
    </div>
  </div>

  </div>
</div>
```

Aquí mostro el tercer bloc que conté info interessant de la pàgina.

```
<!-- Bloc Estadístiques -->
<div class="relative flex flex-col items-start overflow-hidden rounded-3xl bg-[#373737] px-6 pb-6 pt-4 text-left shadow-2xl">
  <h2 class="text-4xl font-serif font-bold text-white tracking-tight sm:text-4xl z-10">
    Estadístiques
  </h2>
  </div>
  <div class="relative mt-6 space-y-4 text-white z-10 w-full">
    <div v-for="stat in newStats" :key="stat.label" class="flex justify-between text-lg border-b border-gray-500 pb-2">
      <span class="font-semibold">{{ stat.label }}</span>
      <span>{{ stat.value }}</span>
    </div>
  </div>
</div>
```

 INSTITUT DE L'EBRE TORTOSA	M7 Desenvolupament d'interfícies	DAM 2n
--	--	------------------

La info del tercer bloc és definida aquí amb un script.

```
<script setup>
const newStats = [
  {label: 'Usuaris Actius', value: '150+'},
  {label: 'Desenvolupament', value: '6 mesos'},
  {label: 'Funcionalitats', value: '10+'},
  {label: 'Compatibilitat', value: '5 navegadors'},
  {label: 'Feedback Positiu', value: '95%'},
  {label: 'Sessions Diàries', value: '500+'},
  {label: 'Projectes Complets', value: '8'},
  {label: 'Temps Mitjà de Resposta', value: '2s'},
  {label: 'Temps de Càrrega', value: '1.5s'}
]
</script>
```

CALCULADORA

PÀGINA

A la pàgina secundària veurem el nav, footer i calculadora.



© 2024 Harvey. All rights reserved.

CODI

El codi consisteix amb varios botons que criden a cada funció.

```
<div class="grid grid-cols-4 gap-2">
  <!-- All X buttons in a single color section -->
  <button @click="applyPercentage" class="bg-red-600 text-white text-2xl p-4 rounded-md hover:bg-red-500">%</button>
  <button @click="clearEntry" class="bg-red-600 text-white text-2xl p-4 rounded-md hover:bg-red-500">CE</button>
  <button @click="clearDisplay" class="bg-red-600 text-white text-2xl p-4 rounded-md hover:bg-red-500">C</button>
  <button @click="deleteDigit" class="bg-red-600 text-white text-2xl p-4 rounded-md hover:bg-red-500">DEL</button>

  <button @click="cube" class="bg-blue-600 text-white text-2xl p-4 rounded-md hover:bg-blue-500">x³</button>
  <button @click="square" class="bg-blue-600 text-white text-2xl p-4 rounded-md hover:bg-blue-500">x²</button>
  <button @click="sqrt" class="bg-blue-600 text-white text-2xl p-4 rounded-md hover:bg-blue-500">√x</button>

  <!-- Dividing button -->
  <button @click="inputOperation( operator: '/' )" class="bg-blue-600 text-white text-2xl p-4 rounded-md hover:bg-blue-500">÷</button>

  <!-- Number buttons -->
  <button @click="inputDigit( digit: '7' )" class="bg-red-700 text-white text-2xl p-4 rounded-md hover:bg-red-600">7</button>
  <button @click="inputDigit( digit: '8' )" class="bg-red-700 text-white text-2xl p-4 rounded-md hover:bg-red-600">8</button>
  <button @click="inputDigit( digit: '9' )" class="bg-red-700 text-white text-2xl p-4 rounded-md hover:bg-red-600">9</button>

  <!-- Multiply button -->
  <button @click="inputOperation( operator: '*' )" class="bg-blue-600 text-white text-2xl p-4 rounded-md hover:bg-blue-500">×</button>
```

Després, aquí veurem cada funció que decideix la funcionalitat dels botons.

```
<script setup>
import { ref } from 'vue';

const displayValue = ref(value: '0');
let lastOperation = false;

Show usages  ⚡ Wharvey123
function clearDisplay() {
  displayValue.value = '0';
  lastOperation = false;
}

Show usages  ⚡ Wharvey123
function clearEntry() {
  if (displayValue.value !== '0') {
    displayValue.value = displayValue.value.slice(0, -1) || '0';
  }
}

Show usages  ⚡ Wharvey123
function inputDigit(digit) {
  if (lastOperation) {
    displayValue.value = digit;
    lastOperation = false;
  } else if (displayValue.value === '0') {
    displayValue.value = digit;
  } else {
    displayValue.value += digit;
  }
}
```

Finalment ajustarem el disseny de la calculadora amb un style.

```
<style scoped>
.web-bg {
  background-image: url('https://media.istoc
  background-size: cover;
  opacity: 0.2;
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  border-radius: 10px;
  z-index: 0;
}

button {
  position: relative;
  z-index: 1;
}
</style>
```

TESTS

El primer test ha de comprovar tota la funcionalitat de la calculadora.

```
RERUN src/tests/calculadora.test.js x7

✓ src/tests/calculadora.test.js (13)
  ✓ Calculadora (13)
    ✓ realitza la suma de 5 + 5
    ✓ realitza la resta de 10 - 5
    ✓ realitza la multiplicació de 5 × 5
    ✓ realitza la divisió de 10 ÷ 2
    ✓ borra la pantalla al prémer C
    ✓ elimina un dígit al prémer DEL
    ✓ realitza la operació de quadrat x²
    ✓ realitza la operació de arrel quadrada √x
    ✓ realitza la operació de percentatge
    ✓ Operació amb decimals
    ✓ Operació combinada
    ✓ realitza la potència de 2^3
    ✓ realitza el cub de 3

Test Files  1 passed (1)
Tests       13 passed (13)
Start at    16:59:37
Duration    157ms
```

Aquí veurem algun exemple del codi que fa els tests.

```
import { mount } from '@vue/test-utils';
import Calculadora from '@views/Calculadora.vue';
import { describe, it, expect } from 'vitest';

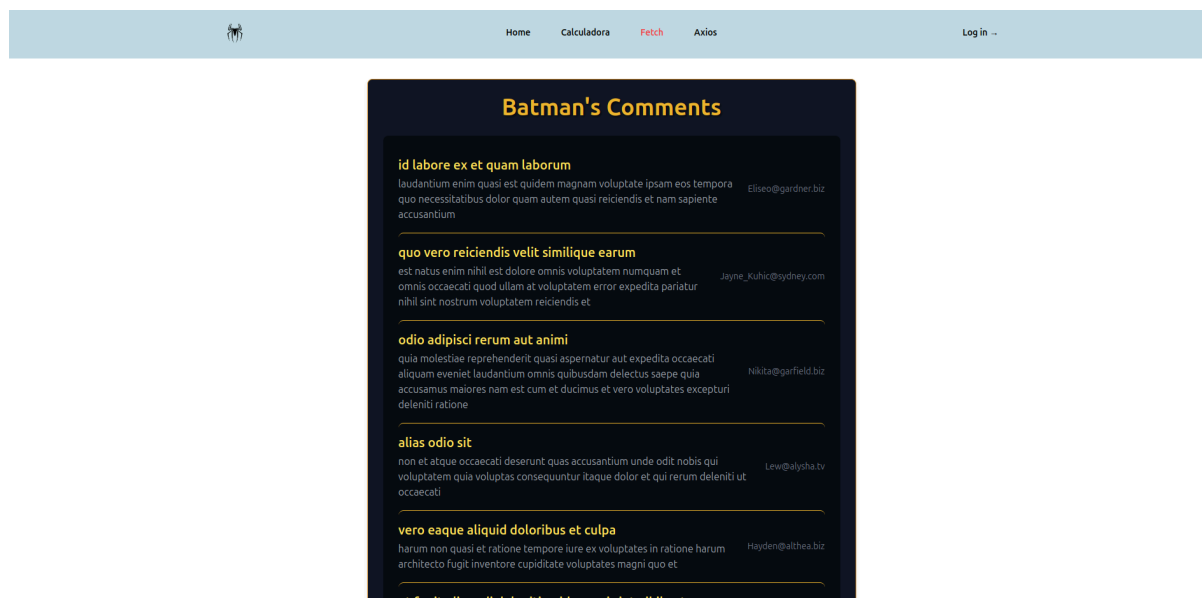
describe('Calculadora', () => {
  it('realitza la suma de 5 + 5', async () => {
    const wrapper: VueWrapper<..., ComponentPublicInstance<...> = mount(Calculadora);
    await wrapper.find(selector: '[data-testid="digit-5"]').trigger(eventString: 'click');
    await wrapper.find(selector: '[data-testid="operation-add"]').trigger(eventString: 'click');
    await wrapper.find(selector: '[data-testid="digit-5"]').trigger(eventString: 'click');
    await wrapper.find(selector: '[data-testid="equal"]').trigger(eventString: 'click');

    expect(wrapper.text()).toContain(item: '10');
  });
});
```


API FETCH

PÀGINA

A la pàgina fetch veurem el nav, footer i info de la api fet amb fetch.



CODI

Amb el fetch capturo la info de la api.

```
<script setup> Show component usages
import { onMounted, ref } from 'vue'

const comments = ref( value: [])
onMounted( hook: () => {
  fetch( input: 'https://jsonplaceholder.typicode.com/comments')
    .then(response => response.json())
    .then(data => {
      comments.value = data
    })
    .catch(error => console.log(error))
})
</script>
```

Amb el següent codi mostro la info i li aplico un estil adequat per a la meva pàgina.

```
<template>
  <div class="pb-4">
    <div class="max-w-3xl mx-auto mt-8 p-6 bg-gray-900 shadow-md rounded-lg border border-yellow-600">
      <h1 class="text-4xl font-bold text-center text-yellow-500 mb-6">Batman's Comments</h1>
      <ul class="divide-y divide-yellow-500">
        <li>
          <div class="py-4 flex flex-col sm:flex-row sm:items-center transition-all duration-300 transform">
            <div class="flex-1">
              <h2 class="text-xl font-semibold text-yellow-300">{{ comment.name }}</h2>
              <p class="text-gray-400 mt-1">{{ comment.body }}</p>
            </div>
            <p class="text-gray-500 text-sm mt-2 sm:mt-0 sm:ml-4">{{ comment.email }}</p>
          </div>
        </li>
      </ul>
    </div>
  </div>
</template>
```

```
<style scoped>
body {
  background-image: url('https://example.com/gotham_background.jpg'); /* Add a Gotham background */
  background-size: cover;
  background-position: center;
}

h1 {
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.7);
}

ul {
  background: rgba(0, 0, 0, 0.5); /* Add semi-transparent background to the list */
  padding: 1rem;
  border-radius: 8px;
}

li {
  transition: background-color 0.3s, transform 0.3s; /* Include transform in the transition */
}

li:hover {
  background-color: rgba(255, 255, 255, 0.1); /* Add hover effect on comments */
  margin-left: 0.5rem; /* Increase left margin on hover */
  margin-right: 0.5rem; /* Increase right margin on hover */
  padding: 1rem; /* Add padding for better spacing */
}
</style>
```

TESTS

El segon test ha de comprovar tot el funcionament de la petició en fetch

```
RERUN src/tests/fetch.test.js x9

✓ src/tests/fetch.test.js (1)
  ✓ loads comments correctly

Test Files  1 passed (1)
Tests       1 passed (1)
Start at    17:27:19
Duration    183ms
```

Aquí veurem algun exemple del codi que fa els tests.

```
import { mount } from '@vue/test-utils';
import { it, expect, beforeEach, vi } from 'vitest';
import Fetch from '@/views/APIFetch.vue'; // Adjust the path to your component

// Mock the fetch API
global.fetch = vi.fn().mockImplementation(() =>
  Promise.resolve({ value: {
    json: () => Promise.resolve({ value: [
      {
        "postId": 1,
        "id": 1,
        "name": "id labore ex et quam laborum",
        "email": "Eliseo@gardner.biz",
        "body": "laudantium enim quasi est quidem magnam voluptate ipsam",
      },
      {
        "postId": 1,
        "id": 2,
        "name": "quo vero reiciendis velit similique earum",
        "email": "Jayne_Kuhic@sydney.com",
        "body": "est natus enim nihil est dolore omnis voluptatem numqua",
      },
    ] }
  })
});
```

API AXIOS

PÀGINA

A la pàgina axios veurem el nav, footer i info de la api fet amb axios.



CODI

Amb el axios capturo la info de la api.

```
<script setup> Show component usages
import { onMounted, ref } from 'vue'
import axios from 'axios'

const users = ref( value: [] ) // Array to hold user data

onMounted( hook: () => {
  axios.get( url: 'https://dummyjson.com/users' )
    .then(response => {
      users.value = response.data.users // Access user
    })
    .catch(error => console.error(error))
})
</script>
```

```
<template>
  <div class="pb-4">
    <div class="max-w-3xl mx-auto mt-8 p-6 bg-blue-900 shadow-md rounded-lg border border-red-600">
      <h1 class="text-4xl font-bold text-center text-yellow-400 mb-6">Avengers Users</h1>
      <ul class="divide-y divide-red-400">
        <li>
          <div class="py-4 flex flex-col sm:flex-row sm:items-center transition-all duration-300 transform hover:scale-105">
            <!-- Profile Image -->
            

            <!-- User Info -->
            <div class="flex-1">
              <h2 class="text-xl font-semibold text-yellow-300">{{ user.firstName }} {{ user.lastName }}</h2>
              <p class="text-gray-200 mt-1">Age: {{ user.age }}</p>
              <p class="text-gray-200 mt-1">Phone: {{ user.phone }}</p>
              <p class="text-gray-200 mt-1">Address: {{ user.address.city }}, {{ user.address.state }}</p>
            </div>

            <!-- Email -->
            <p class="text-gray-400 text-sm mt-2 sm:mt-0 sm:ml-4">{{ user.email }}</p>
          </div>
        </li>
      </ul>
    </div>
  </div>
</template>
```

```
<style scoped>
body {
  background-image: url('https://example.com/avengers_background.jpg');
  background-size: cover;
  background-position: center;
}

h1 {
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.9);
}

ul {
  background: rgba(0, 0, 0, 0.5); /* Semi-transparent background for the list */
  padding: 1rem;
  border-radius: 8px;
}

li {
  transition: background-color 0.3s, transform 0.3s; /* Include transform */
}

li:hover {
  background-color: rgba(255, 255, 255, 0.1); /* Hover effect on comment */
  margin-left: 0.5rem; /* Increase left margin on hover */
  margin-right: 0.5rem; /* Increase right margin on hover */
  padding: 1rem; /* Add padding for better spacing */
}
</style>
```

TESTS

El tercer test ha de comprovar tot el funcionament de la petició en axios

```
DEV v2.1.4 /home/alumnat/Code/javascriptTinkering

✓ src/tests/axios.test.js (1)
  ✓ axios.vue (1)
    ✓ loads users correctly

Test Files  1 passed (1)
Tests       1 passed (1)
Start at    17:28:54
Duration    869ms (transform 174ms, setup 0ms, collect 248ms, tests 125ms, environment 253ms, prepare 129ms)
```

Aquí veurem algun exemple del codi que fa els tests.

```
import { mount } from '@vue/test-utils';
import { describe, it, expect, vi } from 'vitest';
import axios from 'axios';
import axiosVue from '@views/APIAxios.vue';

vi.mock('path: axios');

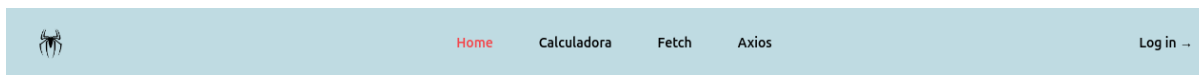
describe('axios.vue', () => {
  it('loads users correctly', async () => {
    const mockUsers : {users: [{firstName: string, lastName:...}] = {
      users: [
        {
          id: 1,
          firstName: "Emily",
          lastName: "Johnson",
          age: 28,
          email: "emily.johnson@x.dummyjson.com",
          phone: "+81 965-431-3024",
          address: {
            city: "Phoenix",
            state: "Arizona"
          },
          image: "https://dummyjson.com/icon/emilys/128"
        },
      ],
    },
  },
}
```



EXTRA

HEADER/NAV

Aquí veurem el nav amb l'opció de pàgina actual que està el usuari il·luminat.



El codi serà així per el header normal:

```
<template> Show component usages
<header class="bg-[#C1DBE3]"> <!-- Header background color -->
  <nav class="mx-auto flex max-w-7xl items-center justify-between p-6 lg:px-8" aria-label="Global">
    <div class="flex lg:flex-1">
      <RouterLink to="/" class="-m-1.5 p-1.5">
        <span class="sr-only">Your Company</span>
        
      </RouterLink>
    </div>
    <div class="flex lg:hidden">
      <button type="button" class="-m-2.5 inline-flex items-center justify-center rounded-md p-2.5 text-black hover:text-red-500">
        <span class="sr-only">Open main menu</span>
        <Bars3Icon class="h-6 w-6" aria-hidden="true" />
      </button>
    </div>
    <div class="hidden lg:flex lg:gap-x-12">
      <RouterLink
        v-for="item in navigation"
        :key="item.name"
        :to="item.href"
        :class="{ 'text-red-500 font-bold': $route.path === item.href, 'text-black': $route.path !== item.href}"
        class="text-sm/6 font-semibold hover:text-red-500"
      >
        {{ item.name }}
      </RouterLink>
    </div>
    <div class="hidden lg:flex lg:flex-1 lg:justify-end">
      <RouterLink
        to="/login"
        :class="{ 'text-red-500 font-bold': $route.path === '/login', 'text-black': $route.path !== '/login'"
        class="text-sm/6 font-semibold hover:text-red-500"
      >
        Log in <span aria-hidden="true">&arr; </span>
      </RouterLink>
    </div>
  </nav>
```


El codi sera així per mode mòvil:


```
<Dialog class="lg:hidden" @close="mobileMenuOpen = false" :open="mobileMenuOpen">
  <div class="fixed inset-0 z-10" />
  <DialogPanel class="fixed top-0 right-0 z-10 w-full bg-[#C1D8E3] px-6 py-6 sm:ring-1 sm:ring-white/10">
    <div class="flex items-center justify-between w-full">
      <RouterLink to="/" class="-m-1.5 p-1.5">
        <span class="sr-only">Your Company</span>
        
      </RouterLink>
      <button type="button" class="-m-2.5 rounded-md p-2.5 text-black hover:text-red-500" @click="mobileMenuOpen = false">
        <span class="sr-only">Close menu</span>
        <XMarkIcon class="h-6 w-6" aria-hidden="true" />
      </button>
    </div>
    <div class="mt-6 flow-root text-center">
      <div class="-my-6 divide-y divide-gray-500/25">
        <div class="space-y-2 py-6">
          <RouterLink
            v-for="item in navigation"
            :key="item.name"
            :to="item.href"
            :class="{ 'text-red-500 font-bold': $route.path === item.href, 'text-black': $route.path !== item.href }"
            class="-mx-3 block rounded-lg px-3 py-2 text-base/7 font-semibold hover:text-red-500 hover:bg-gray-800"
            @click="mobileMenuOpen = false"
          >
            {{ item.name }}
          </RouterLink>
        </div>
        <div class="py-6">
          <RouterLink
            to="/login"
            :class="{ 'text-red-500 font-bold': $route.path === '/login', 'text-black': $route.path !== '/login' }"
            class="-mx-3 block rounded-lg px-3 py-2.5 text-base/7 font-semibold hover:text-red-500 hover:bg-gray-800"
            @click="mobileMenuOpen = false"
          >
            Log in
          </RouterLink>
        </div>
      </div>
    </div>
  </DialogPanel>
```

Les rutes és mostren així.

```
<script setup>
import {ref} from 'vue'
import {Dialog, DialogPanel} from '@headlessui/vue'
import {Bars3Icon, XMarkIcon} from '@heroicons/vue/24/outline'
import {RouterLink, useRoute} from 'vue-router'

const navigation = [
  {name: 'Home', href: '/'},
  {name: 'Calculadora', href: '/calculadora'},
  {name: 'Fetch', href: '/api-fetch'},
  {name: 'Axios', href: '/api-axios'},
]

const mobileMenuOpen = ref({ value: false })
</script>
```

 INSTITUT DE L'EBRE TORTOSA	M7 Desenvolupament d'interfícies	DAM 2n
--	--	------------------

FOOTER

Aquí veurem el footer simple amb el meu nom mostrant els drets reservats.

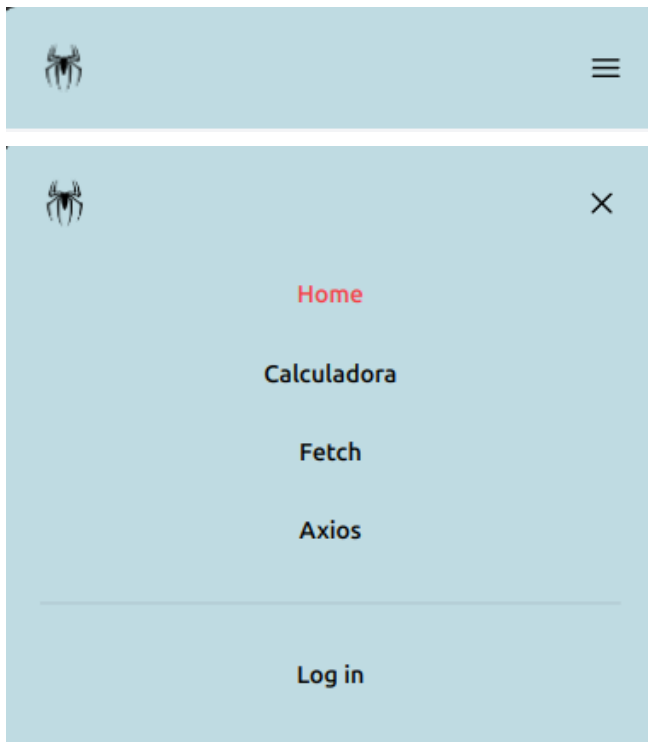
© 2024 Harvey. All rights reserved.

Codi del footer.

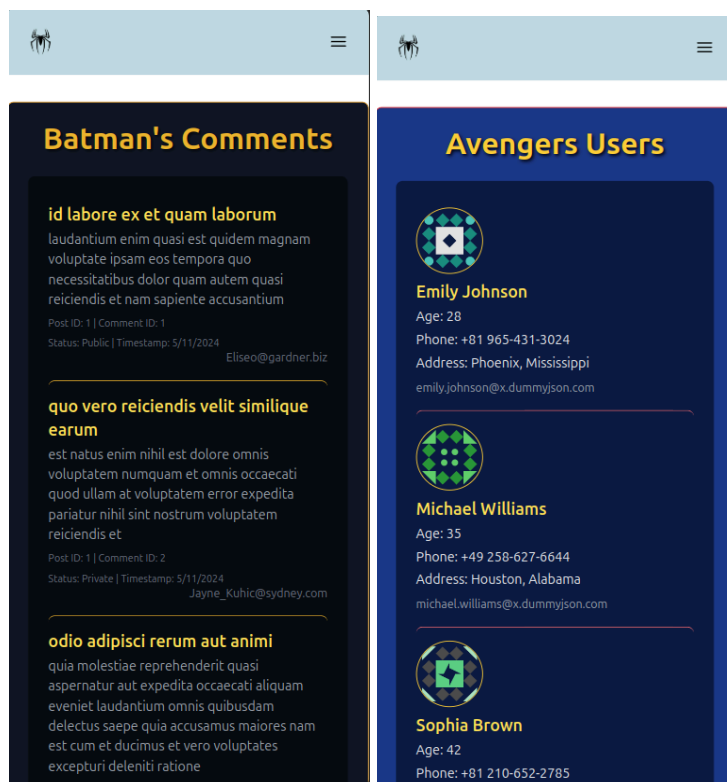
```
<template> Show component usages
<footer class="bg-[#C1DBE3]"> <!-- Color de fons -->
  <div class="mx-auto max-w-5xl px-5 py-7 md:flex md:items-center md:justify-center lg:px-8">
    <p class="text-center font-bold text-xl font-serif text-[#FF0000]"> <!-- Augmenta la mida -->
      &copy; 2024 Harvey. All rights reserved.
    </p>
  </div>
</footer>
</template>
```

RESPONSIVE

El menu responsive quedaria així:



Les pàgines principals en modo responsive quedarien així.



SERVIDOR

NETLIFY

En aquest cas he muntat el projecte a netlify.

LINK: <https://vueharvey.netlify.app/>

