



Exam 4 - Part 1: Pet Adoption Center Project

Instructions:

Today you will build a small Vue 3 project using what you've learned so far.

This is an **open book, open notes, open conversation lab**.

At the end of class, **you must push your completed project and your notes to GitHub**.

- ✓ You are encouraged to ask questions, help each other, and use your notes.
- ✓ Bootstrap is optional for styling.



Project Overview:

You are creating a **Pet Adoption Center** with:

- A **Home page** that welcomes users
- A **Pets page** that displays a list of pets
- A **Nav bar** that lets users navigate between pages
- A **Search bar** where users can type a pet's name and see if it is available for adoption
- **Props** passed to components
- An **Event Emit** from the Nav bar to App.vue
- **Simple Router setup** (no dynamic parameters)



Step-by-Step Instructions:

1. Create a new Vue project:

```
npm create vue@latest
```

Give it a project name of vue-exam4

✓ Choose:

- TypeScript: NO
- JSX Support NO
- Router: YES
- Pinia: NO
- Vitest: NO
- End-to_end Testing NO
- ESLint (error prevention) NO

- Prettier NO
-

2. Set up the basic file structure:

In your `src/` folder:

```
src/
  components/
    Nav.vue
    PetCard.vue
  views/
    Home.vue
    PetList.vue
  App.vue
  main.js
  router/
    index.js (already created)
```

3. Router Configuration (`router/index.js`):

✅ Make sure you only have **two routes**:

```
import { createRouter, createWebHistory } from 'vue-router'
import Home from '../views/Home.vue'
import PetList from '../views/PetList.vue'

const routes = [
  { path: '/', component: Home },
  { path: '/pets', component: PetList }
]

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes
})

export default router
```

4. App.vue

✅ Your App.vue will:

- Import Nav.vue
- Hold the **pets array** and **search result** message
- Pass **props** (pets) down to router-view
- Handle search logic from Nav.vue

Example outline:

```
1  <script>
2  import Nav from './components/Nav.vue'
3
4  export default {
5    components: { Nav },
6    data() {
7      return {
8        pets: [
9          { id: 1, name: "Buddy", type: "Dog", age: 2, adopted: true },
10         { id: 2, name: "Mittens", type: "Cat", age: 4, adopted: false },
11         { id: 3, name: "Tweety", type: "Bird", age: 1, adopted: true },
12         { id: 4, name: "Shadow", type: "Dog", age: 5, adopted: false }
13       ],
14       searchResult: ''
15     },
16     methods: {
17       handleSearch(name) {
18         const pet = this.pets.find(pet => pet.name.toLowerCase() === name.
19           toLowerCase());
20         if (pet) {
21           this.searchResult = `${pet.name} is ${pet.adopted ? "adopted" :
22             "available"}`;
23         } else {
24           this.searchResult = `No pet found by the name ${name}.`;
25         }
26       }
27     }
28   }
29
30   </script>
31
32   <template>
33     <div>
34       <Nav @search-pet="handleSearch" />
35       <p>{{ searchResult }}</p>
36       <router-view :pets="pets" />
37     </div>
38   </template>
```

5. Nav.vue

✓ Create a navigation bar with:

- Two router-links (Home and Pets)
- A search input and button that emits a search event

Example:

```

1  <script>
2  export default {
3    data() {
4      return {
5        searchName: ''
6      }
7    },
8    methods: {
9      search() {
10       this.$emit('search-pet', this.searchName);
11     }
12   }
13 }
14 </script>
15
16 <template>
17   <nav>
18     <router-link to="/">Home</router-link> |
19     <router-link to="/pets">View Pets</router-link>
20
21     <div>
22       <input v-model="searchName" type="text" placeholder="Enter pet name" />
23       <button @click="search">Search</button>
24     </div>
25   </nav>
26 </template>
27

```

6. Home.vue

✓ Simple welcome page:

```

<template>
  <div>
    <h1>Welcome to the Pet Adoption Center!</h1>
    <p>Find your next best friend today!</p>
  </div>
</template>

```

7. PetList.vue

✓ Displays all pets passed as a prop:

```

<script>
import PetCard from '../components/PetCard.vue'

export default {
  components: { PetCard },
  props: ['pets']
}
</script>

<template>
  <div>
    <h2>Available Pets:</h2>
    <div v-for="pet in pets" :key="pet.id">
      <PetCard :pet="pet" />
    </div>
  </div>
</template>

```

8. PetCard.vue

✓ Displays individual pet information:

```

<script>
export default {
  props: ['pet']
}
</script>

<template>
  <div class="card">
    <h3>{{ pet.name }}</h3>
    <p>Type: {{ pet.type }}</p>
    <p>Age: {{ pet.age }} years</p>
    <p v-if="pet.adopted">Status: Adopted</p>
    <p v-else>Status: Available</p>
  </div>
</template>

```



Final Reminders:

- ✓ Focus on simple functionality.
- ✓ Open book and notes are allowed today.
- ✓ Push your final project and notes to GitHub by the end of class.
- ✓ Prepare for Exam 4 Part 2 next class, where only your GitHub notes can be used.



In simple words:

"Build a small pet adoption app with two pages, a search bar, and display pets using props and emits."

