**Vue.js and CodeIgniter GUIDE**

Part 1: Setting Up Vue.js Frontend

Step 1: Install Vue CLI

If you don't have Vue CLI installed, open your terminal and run the following command:

|  |
| --- |
| npm install -g @vue/cli |

Step 2: Create a Vue Project

Create a new Vue project using the following commands:

|  |
| --- |
| vue create your-vue-project  cd your-vue-project |

Follow the prompts to set up your project.

Step 3: Test Your Vue Project

Run your Vue project to ensure it's set up correctly:

|  |
| --- |
| npm run serve |

Visit `http://localhost:8080` in your browser. You should see the Vue default page.

Part 2: Setting Up CodeIgniter Backend

Step 1: Install CodeIgniter

If you haven't installed CodeIgniter, you can use Composer to create a new project:

|  |
| --- |
| composer create-project codeigniter4/appstarter your-codeigniter-project  cd your-codeigniter-project |

Step 2: Set Up Database

Certainly! Setting up the database in CodeIgniter involves configuring the database settings and running migrations to create the necessary tables. Let's expand on this step:

Step 3: Create API Endpoints

Create controllers and methods to handle API requests. For example, you might create a controller named `Api` with methods like `getUsers`, `createUser`, etc.

Step 4: Enable CORS

Install a library like `myth:app` to handle Cross-Origin Resource Sharing (CORS). Run the following command:

|  |
| --- |
| composer require myth/auth |

Follow the documentation to configure CORS.

Part 3: Connecting Vue.js to CodeIgniter Backend

Step 1: Install Axios

Axios is a popular HTTP client for making API requests. Install it in your Vue project:

|  |
| --- |
| npm install axios |

Step 2: Create API Service

Create a service file in your Vue project to handle API requests. For example, create a file named `apiService.js`:

|  |
| --- |
| // apiService.js  import axios from 'axios';  const baseURL = 'http://localhost:8080'; // Replace with your CodeIgniter backend URL  const api = axios.create({  baseURL,  });  export const getUsers = () => api.get('/api/getUsers');  export const createUser = (userData) => api.post('/api/createUser', userData);  // Add more methods as needed |

Step 3: Use API Service in Vue Components

Now, you can use the created API service in your Vue components:

|  |
| --- |
| <!-- ExampleComponent.vue -->  <template>  <div>  <h1>Users</h1>  <ul>  <li v-for="user in users" :key="user.id">{{ user.name }}</li>  </ul>  </div>  </template>  <script>  import { getUsers } from '@/services/apiService';  export default {  data() {  return {  users: [],  };  },  mounted() {  getUsers()  .then((response) => {  this.users = response.data;  })  .catch((error) => {  console.error('Error fetching users:', error);  });  },  };  </script> |