

Mon Application de TodoList avec Vue.js 3

NDOH NDOH Jules Moïse Yannick

ndohmoise@gmail.com | +237 6 95 99 40 53

29 octobre 2024

Table des matières

1	Projet 1 : My TodoList	2
1.1	Présentation du projet	2
1.2	Installation des Bibliothèques nécessaires	2
1.2.1	Installation du Projet	2
1.2.2	Installation du Routeur	3
1.2.3	Installation de bootstrap	3
1.2.4	Installation de axios	3
1.2.5	Installatino de sweetalert2	3
1.2.6	Création du fichier .env	4
1.3	Mise en place du Router et des Components	4
1.3.1	Les Components	4
1.3.2	Mise à jour du fichier main.js	5
1.3.3	Mise à jour du fichier MyTodoList.vue	6
1.3.4	Mise à jour du fichier LoginPage.vue	8
1.3.5	Mise à jour du fichier RegisterPage.vue	11
1.3.6	Mise à jour du fichier ProjectEdit.vue	15
1.3.7	Mise à jour du fichier ProjectList.vue	18
1.3.8	Mise à jour du fichier ProjectShow.vue	21
1.3.9	Mise à jour du fichier Footer.vue	23
1.3.10	Mise à jour du fichier Header.vue	23
1.3.11	Les résultats Obtenus	26

Chapitre 1

Projet 1 : My TodoList

1.1 Présentation du projet

Bonjour, aujourd'hui nous allons voir comment développer la connexion et l'enregistrement dans Vue.js en utilisant une API. La plupart des applications web aujourd'hui qui ont des utilisateurs ou différents types d'utilisateurs nécessitent une connexion ou une authentification pour protéger les fonctionnalités et les informations de données du système. Et aujourd'hui, je vais vous montrer comment faire une connexion utilisateur et un enregistrement dans Vue.js.

Vue JS également appelé vue.js est une bibliothèque JavaScript open source utilisée pour construire des interfaces utilisateur (UI). C'est l'une des bibliothèques JavaScript les plus populaires pour construire l'avant-garde.

Dans ce tutoriel, nous utiliserons l'API disponible qui est fabriquée par Binaryboxtus. Vous pouvez le vérifier sur le lien suivant : [lien](#) Notre projet contiendra les pages suivantes :

1. une page de connexion
2. une page d'inscription
3. la page principale de travail qui contient les pages suivantes :
 - Une page de création des tâches
 - Une page de Mise à jour des tâches
 - Une page d'affichage des tâches

1.2 Installation des Bibliothèques nécessaires

1.2.1 Installation du Projet

Nous allons démarrer un nouveau projet avec le code suivant :

```
npm create vite@latest
```

1.2.2 Installation du Routeur

Un routeur est un composant qui va permettre de gérer les routes de notre application. Une route est une URL qui va permettre d'accéder à une page de notre application. Nous téléchargeons la dépendance vue-router en version 4.

Pour cela nous allons utiliser le code suivant :

```
npm install vue-router@4
```

1.2.3 Installation de bootstrap

Bootstrap est un framework css qui va nous permettre de faire le design de nos pages.

Pour cela nous allons utiliser le code suivant :

```
npm install bootstrap
```

1.2.4 Installation de axios

Axios est une bibliothèque HTTP basée sur les promesses qui est utilisé pour faire des demandes ou requêtes à un serveur ou à une API.

Pour cela nous allons utiliser le code suivant :

```
npm install axios
```

1.2.5 Installatino de sweetalert2

Sweetalert2 est une bibliothèque qui permet de gérer l'affichage des boites pop-up qui se comporte comme des modal bootstrap.

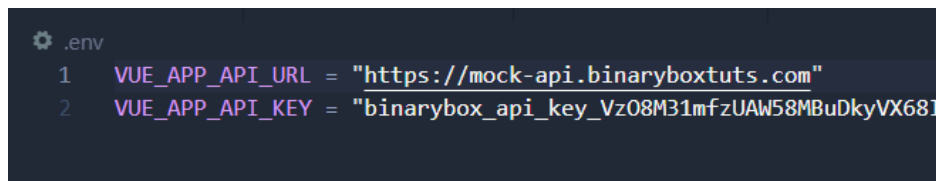
Pour cela nous allons utiliser le code suivant :

```
npm install sweetalert2
```

1.2.6 Création du fichier .env

Après avoir installé les paquets, nous allons maintenant créer notre fichier **.env** sur notre répertoire racine. Nous déclarons l'URL de base de l'API et la clé API dans le fichier **.env**. N'oublions pas de l'ajouter au fichier **.gitignore**.

Pour obtenir une clé pour notre application vous pouvez cliquer sur le lien suivant : [lien](#)

A screenshot of a code editor showing the content of a file named .env. The file contains two lines of text: 1 VUE_APP_API_URL = "https://mock-api.binaryboxtuts.com" and 2 VUE_APP_API_KEY = "binarybox_api_key_Vz08M31mfzUAW58MBuDkyVX68I". The lines are numbered 1 and 2 on the left. The background is dark with light-colored text.

```
.env
1  VUE_APP_API_URL = "https://mock-api.binaryboxtuts.com"
2  VUE_APP_API_KEY = "binarybox_api_key_Vz08M31mfzUAW58MBuDkyVX68I"
```

FIGURE 1.1 – Contenu du fichier .env

1.3 Mise en place du Router et des Composants

1.3.1 Les Components

Nous allons créer 4 fichiers qui seront contenu dans le dossier **pages** :

1. **LoginPage.vue** : qui sera la page d'accueil et de connexion
2. **RegisterPage.vue** : qui sera la page d'inscription
3. **MyTodoList.vue** : qui est la page principale de notre application qui va créer les différentes tâches.
4. **ProjectShow.vue** : qui permet d'afficher les détails d'une tâche.
5. **ProjectList.vue** : qui permet de lister toutes les différentes tâches
6. **ProjectEdit.vue** : qui permet d'afficher les détails d'une tâche pour être modifier.
7. **ProjectCreate.vue** : qui permet de créer une nouvelle tâche.
8. **Header.vue** : Me permet d'afficher le header dans les différentes pages
9. **Footer.vue** : Me permet d'afficher le footer dans les différentes pages.
10. : **LayoutDiv.vue** : Création du component qui sera dans le dossier Component.

1.3.2 Mise à jour du fichier main.js

Dans ce fichier nous allons importer les fichiers nécessaires ainsi que les routes de notre application.

Nous avons le code suivant :

```
import { createApp } from 'vue';
import App from './App.vue';
import axios from 'axios';
import 'bootstrap/dist/css/bootstrap.css';
import { createRouter, createWebHistory } from 'vue-router';
import LoginPage from './components/pages/LoginPage.vue';
import RegisterPage from './components/pages/RegisterPage.vue';
import MyToDoList from './components/pages/MyToDoList.vue';
import ProfilPage from './components/pages/ProfilPage.vue';
import ProjectList from '../src/components/pages/ProjectList.vue';
import ProjectCreate from '../src/components/pages/ProjectCreate.vue';
import ProjectEdit from './components/pages/ProjectEdit.vue';
import ProjectShow from './components/pages/ProjectShow.vue';

axios.defaults.baseURL = "https://mock-api.binaryboxtuts.com"
axios.interceptors.request.use(function (config) {
  config.headers['X-Binarybox-Api-Key'] = "binarybox_api_key_Vz08M3
1mfzUAW58MBuDkyVX68I
WufWJWW7m5BqqSi3FSXHHwKeHjuXQz0C
7QdACzffplQ93npFb6Q3sMQLeImXx
kz3IHQDkWy1yt";
  return config;
});

const router = createRouter({
  history: createWebHistory(),
  routes: [
    { path: '/', component: LoginPage },
    { path: '/register', component: RegisterPage },
    { path: '/mytodolist', component: ProjectList },
    {path: '/profil', component: ProfilPage},
    {path: '/projectlist', component: ProjectList},
    {path: '/create', component: ProjectCreate},
    {path: '/edit/:id', component: ProjectEdit},
    {path: '/show/:id', component: ProjectShow}
```

```

    ],
  });

  createApp(App).use(router).mount('#app');

```

1.3.3 Mise à jour du fichier MyTodoList.vue

Nous avons le code suivant :

```

<template>
<layout-div>
<Header />
</layout-div>
<layout-div>
<h2 class="text-center mt-5 mb-3">Créer une tâche</h2>
<div class="card">
<div class="card-header">
<router-link
class="btn btn-outline-info float-right"
to="/">Voir toutes les tâches
</router-link>
</div>
<div class="card-body">
<form>
<div class="form-group">
<label htmlFor="name">Nom</label>
<input
v-model="project.name"
type="text"
class="form-control"
id="name"
name="name"/>
</div>
<div class="form-group">
<label htmlFor="description">Description</label>
<textarea
v-model="project.description"
class="form-control"
id="description"
rows="3"

```

```

name="description"></textarea>
</div>
<button
@click="handleSave()"
:disabled="isSaving"
type="button"
class="btn btn-outline-primary mt-3">
Enregistrer la tâche
</button>
</form>
</div>
</div>
</layout-div>
<layout-div>
<Footer />
</layout-div>
</template>

<script>
import axios from 'axios';
import LayoutDiv from '../LayoutDiv.vue';
import Swal from 'sweetalert2'
import Header from '../Header.vue'
import Footer from '../Footer.vue'

export default {
  name: 'ProjectCreate',
  components: {
    LayoutDiv,
    Header,
    Footer
  },
  data() {
    return {
      project: {
        name: '',
        description: '',
      },
      isSaving:false,
    };
  },

```



```

methods: {
  handleSave() {
    this.isSaving = true
    axios.post('/api/projects', this.project)
      .then(response => {
        Swal.fire({
          icon: 'succès',
          title: 'Tâche enregistré avec succès!',
          showConfirmButton: false,
          timer: 1500
        })
        this.isSaving = false
        this.project.name = ""
        this.project.description = ""
        return response
      })
      .catch(error => {
        this.isSaving = false
        Swal.fire({
          icon: 'Erreur',
          title: 'Une Erreur est survenue!',
          showConfirmButton: false,
          timer: 1500
        })
        return error
      });
  },
},
};
</script>

```

1.3.4 Mise à jour du fichier LoginPage.vue

Nous avons le code suivant :

```

<script>
import axios from 'axios';
import LayoutDiv from '../LayoutDiv.vue';

export default {

```

```

name: 'LoginPage',
components: {
  LayoutDiv,
},
data() {
  return {
    email:'',
    password:'',
    validationErrors:{},
    isSubmitting:false,
  };
},
created() {
  if(localStorage.getItem('token') !== "" &&
    localStorage.getItem('token') !== null){
    this.$router.push('/mytodolist')
  }
},
methods: {
  loginAction(){
    this.isSubmitting = true
    let payload = {
      email: this.email,
      password: this.password,
    }
    axios.post('/api/login', payload)
      .then(response => {
        localStorage.setItem('token', response.data.token)
        this.$router.push('/mytodolist')
        return response
      })
      .catch(error => {
        this.isSubmitting = false
        if (error.response.data.errors !== undefined) {
          this.validationErrors = error.response.data.errors
        }
        if (error.response.data.error !== undefined) {
          this.validationErrors = error.response.data.error
        }
        return error
      });
  }
}

```

```

    }
  },
};
</script>

<template>
<layout-div>
<div class="row justify-content-md-center mt-5">
<div class="col-4">
<div class="card">
<div class="card-body">
<h5 class="card-title mb-4">Sign In</h5>
<form>
<p v-if="Object.keys(validationErrors).length != 0"
class='text-center '><small class='text-danger'>Incorrect Email or Password</s
<div class="mb-3">
<label
htmlFor="email"
class="form-label">
Email address
</label>
<input
v-model="email"
type="email"
class="form-control"
id="email"
name="email"
/>
</div>
<div class="mb-3">
<label
htmlFor="password"
class="form-label">Password
</label>
<input
v-model="password"
type="password"
class="form-control"
id="password"
name="password"
/>

```



```

    };
  },
  created() {
    if(localStorage.getItem('token') != "" &&
      localStorage.getItem('token') != null){
      this.$router.push('/mytodolist')
    }
  },
  methods: {
    async registerAction(){
      this.isSubmitting = true
      let payload = {
        name:this.name,
        email: this.email,
        password: this.password,
        password_confirmation: this.confirmPassword
      }
      axios.post('/api/register', payload)
        .then(response => {
          localStorage.setItem('token', response.data.token)
          this.$router.push('/mytodolist')
          return response
        })
        .catch(error => {
          this.isSubmitting = false
          if (error.response.data.errors != undefined) {
            this.validationErrors = error.response.data.errors
          }
          return error
        });
    }
  },
};
</script>

```

```

<template>
<layout-div>
<div class="row justify-content-md-center mt-5">
<div class="col-4">
<div class="card">
<div class="card-body">

```

```

<h5 class="card-title mb-4"><b>Inscription</b></h5>
<form >
<div class="mb-3">
<label
htmlFor="name"
class="form-label">Nom
</label>
<input
type="text"
class="form-control"
id="name"
name="name"
v-model="name"
/>
<div v-if="validationErrors.name" class="flex flex-col">
<small class="text-danger">
{{validationErrors?.name[0]}}
</small >
</div>
</div>
<div class="mb-3">
<label
htmlFor="email"
class="form-label">Email
</label>
<input
type="email"
class="form-control"
id="email"
name="email"
v-model="email"
/>
<div v-if="validationErrors.email" class="flex flex-col">
<small class="text-danger">
{{validationErrors?.email[0]}}
</small >
</div>
</div>
<div class="mb-3">
<label
htmlFor="password"

```

```

class="form-label">Mot de passe
</label>
<input
type="password"
class="form-control"
id="password"
name="password"
v-model="password"
/>
<div v-if="validationErrors.password" class="flex flex-col">
<small class="text-danger">
{{validationErrors?.password[0]}}
</small >
</div>
</div>
<div class="mb-3">
<label
htmlFor="confirm_password"
class="form-label">Confirmer Mot de Passe
</label>
<input
type="password"
class="form-control"
id="confirm_password"
name="confirm_password"
v-model="confirmPassword"
/>
</div>
<div class="d-grid gap-2">
<button
:disabled="isSubmitting"
@click="registerAction()"
type="button"
class="btn btn-danger btn-block">Inscription
</button>
<p class="text-center">Vous avez déjà un compte ?
  <router-link to="/">Connecte toi ici</router-link>
</p>
</div>
</form>
</div>

```

```

</div>
</div>
</div>
</layout-div>
</template>

```

1.3.6 Mise à jour du fichier ProjectEdit.vue

Nous avons le code suivant :

```

<template>
<layout-div>
<Header />
</layout-div>
<layout-div>
<h2 class="text-center mt-5 mb-3">Editer La tâche</h2>
<div class="card">
<div class="card-header">
<router-link
class="btn btn-outline-info float-right"
to="/">Voir les tâches
</router-link>
</div>
<div class="card-body">
<form>
<div class="form-group">
<label htmlFor="name">Nom</label>
<input
v-model="project.name"
type="text"
class="form-control"
id="name"
name="name"/>
</div>
<div class="form-group">
<label htmlFor="description">Description</label>
<textarea
v-model="project.description"
class="form-control"
id="description"

```



```

rows="3"
name="description"></textarea>
</div>
<button
@click="handleSave()"
:disabled="isSaving"
type="button"
class="btn btn-outline-primary mt-3">
Enregistrer la tâche
</button>
</form>
</div>
</div>
</layout-div>
<layout-div>
<Header />
</layout-div>
</template>

<script>
import axios from 'axios';
import LayoutDiv from '../LayoutDiv.vue';
import Swal from 'sweetalert2'
import Header from '../Header.vue'
import Footer from '../Footer.vue'

export default {
  name: 'ProjectEdit',
  components: {
    LayoutDiv,
    Header,
    Footer
  },
  data() {
    return {
      project: {
        name: '',
        description: '',
      },
      isSaving: false,
    };
  }
}

```

```

},
created() {
  const id = this.$route.params.id;
  axios.get(`/api/projects/${id}`)
    .then(response => {
      let projectInfo = response.data
      this.project.name = projectInfo.name
      this.project.description = projectInfo.description
      return response
    })
    .catch(error => {
      Swal.fire({
        icon: 'Erreur',
        title: 'Une Erreur est survenue!',
        showConfirmButton: false,
        timer: 1500
      })
      return error
    })
},
methods: {
  handleSave() {
    this.isSaving = true
    const id = this.$route.params.id;
    axios.patch(`/api/projects/${id}`, this.project)
      .then(response => {
        Swal.fire({
          icon: 'succès',
          title: 'La tâche est enregistré avec succès!',
          showConfirmButton: false,
          timer: 1500
        })
        this.isSaving = false
        this.project.name = ""
        this.project.description = ""
        return response
      })
      .catch(error => {
        this.isSaving = false
        Swal.fire({
          icon: 'Erreur',

```

```

        title: 'Une Erreur est survenue!',
        showConfirmButton: false,
        timer: 1500
      })
      return error
    });
  },
},
};
</script>

```

1.3.7 Mise à jour du fichier ProjectList.vue

Nous avons le code suivant :

```

<script>
import axios from 'axios'
import LayoutDiv from '../LayoutDiv.vue'
import Swal from 'sweetalert2'
import Entete from '../Header.vue'
import Footer from '../Footer.vue'

export default {
  name: 'ProjectList',
  components: {
    LayoutDiv,
    Entete,
    Footer
  },
  data() {
    return {
      projects: []
    };
  },
  created() {
    this.fetchProjectList();
  },
  methods: {
    fetchProjectList() {
      axios.get('/api/projects')
    }
  }
}

```

```

.then(response => {
  this.projects = response.data;
  return response
})
.catch(error => {
  return error
});
},
handleDelete(id){
  Swal.fire({
    title: 'êtes-vous sur ?',
    text: "Vous êtes sur le point de supprimer une Tâche",
    icon: 'Avertissement',
    showCancelButton: true,
    confirmButtonColor: '#3085d6',
    cancelButtonColor: '#d33',
    confirmButtonText: 'Oui !'
  }).then((result) => {
    if (result.isConfirmed) {
      axios.delete(`/api/projects/${id}`)
        .then( response => {
          Swal.fire({
            icon: 'Succès',
            title: 'Tâche supprimée avec succès!',
            showConfirmButton: false,
            timer: 1500
          })
          this.fetchProjectList();
          return response
        })
        .catch(error =>{
          Swal.fire({
            icon: 'Erreur',
            title: 'Une erreur est survenue!',
            showConfirmButton: false,
            timer: 1500
          })
          return error
        });
    }
  })
}
}
}

```

```

    }
  },
};
</script>

<template>
<layout-div>
<Entete/>
</layout-div>
<layout-div>
<div class="container">
<br><br>
<div class="card">
<div class="card-header">
<router-link to="/create"
class="btn btn-primary"
>Créer une tâches
</router-link>
</div>
<div class="card-body">

<table class="table table-bordered">
<thead>
<tr>
<th>Nom</th>
<th>Description</th>
<th width="380px">Action</th>
</tr>
</thead>
<tbody>

<tr v-for="project in projects" :key="project.id">
<td>{{project.name}}</td>
<td>{{project.description}}</td>
<td>
<router-link :to="'/show/${project.id}'"
class="btn btn-danger mx-1">Afficher</router-link>
<router-link :to="'/edit/${project.id}'"
class="btn btn-success mx-1">Editer</router-link>
<button
@click="handleDelete(project.id)"

```



```

<script>
import axios from 'axios';
import LayoutDiv from '../LayoutDiv.vue';
import Swal from 'sweetalert2'

export default {
  name: 'ProjectShow',
  components: {
    LayoutDiv,
  },
  data() {
    return {
      project: {
        name: '',
        description: '',
      },
      isSaving:false,
    };
  },
  created() {
    const id = this.$route.params.id;
    axios.get(`/api/projects/${id}`)
      .then(response => {
        let projectInfo = response.data
        this.project.name = projectInfo.name
        this.project.description = projectInfo.description
        return response
      })
      .catch(error => {
        Swal.fire({
          icon: 'Erreur',
          title: 'Une Erreur est survenue!',
          showConfirmButton: false,
          timer: 1500
        })
        return error
      })
  },
  methods: {

```

```
  },  
};  
</script>
```

1.3.9 Mise à jour du fichier Footer.vue

Nous avons le code suivant :

```
<template>  
<footer>  
<div id="pieds">  
My TodoList@Vue.Js3  
</div>  
</footer>  
</template>  
  
<script setup>  
  
</script>  
  
<style lang="scss" scoped>  
footer{  
  position: relative;  
  bottom: -200px;  
}  
#pieds{  
  width: 100%;  
  bottom : 0px;  
  color: white;  
  background-color: black;  
  text-align: center;  
  padding: 10px;  
  font-weight: bold;  
}  
</style>
```

1.3.10 Mise à jour du fichier Header.vue

```
<script>
```



```

import axios from 'axios';
import LayoutDiv from './LayoutDiv.vue'

export default {
  name: 'MyTodoList',
  components: {
    LayoutDiv,
  },
  data() {
    return {
      user: {},
    };
  },
  created() {
    this.getUser();
    if(localStorage.getItem('token') == "" ||
      localStorage.getItem('token') == null){
      this.$router.push('/')
    }else {
      this.getUser();
    }
  },
  methods: {
    getUser() {
      axios.get('/api/user', { headers:{Authorization:
        'Bearer ' + localStorage.getItem('token')}})
        .then((r) => {
          this.user = r.data;
          return r
        })
        .catch((e) => {
          return e
        });
    },

    logoutAction () {
      axios.post('/api/logout',{}, { headers:{Authorization:
        'Bearer ' + localStorage.getItem('token')}})
        .then((r) => {
          localStorage.setItem('token', "")

```

```

        this.$router.push('/')
        return r
      })
      .catch((e) => {
        return e
      });
    }

  },
};
</script>

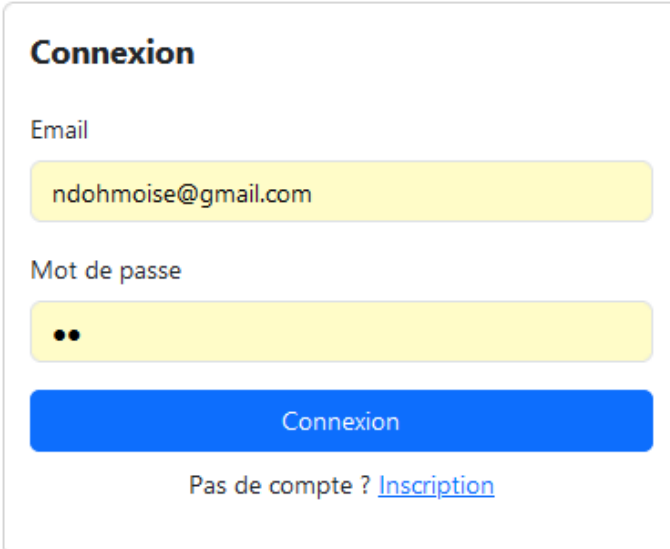
<template>
<layout-div>
<div class="row justify-content-md-center">
<div class="col-12">
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="#"><b>MyToDoList : Tableau de Bord</b></a>
<div class="d-flex">
<ul class="navbar-nav">
<li class="nav-item">
<a @click="logoutAction()" class="nav-link" aria-current="page"
  href="#"><b>Deconnexion</b></a>
</li>
</ul>
</div>
</div>
</nav>
<h2 class="text-center mt-5">Bienvenue, {{user?.name}}!</h2  >
</div>
</div>
</layout-div>
</template>

```

Nous avons le lien suivant pour les résultats :

1. github : lien github
2. vercel : lien vercel

1.3.11 Les résultats Obtenus



Connexion

Email

ndohmoise@gmail.com

Mot de passe

••

Connexion

Pas de compte ? [Inscription](#)

FIGURE 1.2 – Page d'accueil

Inscription

Nom

Email

ndohmoise@gmail.com

Mot de passe

••

Confirmer Mot de Passe

Inscription

Vous avez déjà un compte ? [Connecte toi ici](#)

FIGURE 1.3 – Page d’inscription

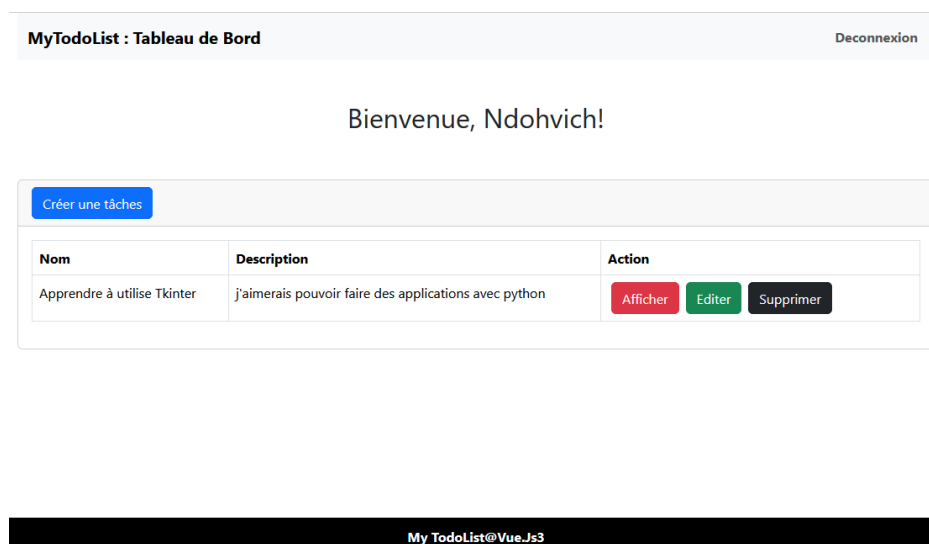


FIGURE 1.4 – Page d’administration centrale