Home Page

About Page (Not logged in)

About Page (After logged in)

router/index.js

```js
import { createRouter, createWebHistory } from 'vue-router'

import HomeView from '../views/HomeView.vue'

import AboutView from '../views/AboutView.vue'

import LoginView from '../views/LoginView.vue'

import AccessDeniedView from '@/views/AccessDeniedView.vue'


const routes = [
 {
  path: '/',
  name: 'Home',
  component: HomeView
 },
 {
  path: '/About',
  name: 'About',
  component: AboutView,
  beforeEnter: (to, from, next) => {
   const isAuthenticated = localStorage.getItem('isAuthenticated')
   if (isAuthenticated === 'true') {
    next()
   } else {
    next('/AccessDenied')
   }
  }
 },
 {
  path: '/Login',
```

```javascript
    name: 'Login',

    component: LoginView

   },

   {

    path: '/AccessDenied',

    name: 'AccessDenied',

    component: AccessDeniedView

   }

]


const router = createRouter({

  history: createWebHistory(),

  routes

})


export default router
```

src/App.vue

```
<script setup>

import JSONLab from './components/JSONLab.vue'

import BHeader from './components/BHeader.vue'

import LibraryRegistrationForm from './components/LibraryRegistrationForm.vue'

</script>


<template>

 <header>

  <BHeader />

 </header>


 <main>

  <!-- <LibraryRegistrationForm /> -->

  <!-- <JSONLab /> -->

  <router-view></router-view>

 </main>

</template>
```

src/views/HomeView.vue

```
<script setup>

import { ref } from 'vue'

import DataTable from 'primevue/datatable'

import Column from 'primevue/column'


const formData = ref({

 username: '',

 password: '',

 confirmPassword: '',

 isAustralian: false,

 reason: '',

 gender: '',

 suburb: 'Clayton'

})


const submittedCards = ref([])


const submitForm = () => {

 validateName(true)

 validatePassword(true)

 if (!errors.value.username && !errors.value.password) {

  submittedCards.value.push({ ...formData.value })

  clearForm()

 }

}


const clearForm = () => {
```

```
  formData.value = {
   username: '',
   password: '',
   isAustralian: false,
   reason: '',
   gender: ''
 }
}

const errors = ref({
 username: null,
 password: null,
 confirmPassword: '',
 resident: null,
 gender: null,
 reason: null,
 suburb: null
})

const validateName = (blur) => {
 if (formData.value.username.length < 3) {
  if (blur) errors.value.username = 'Name must be at least 3 characters'
 } else {
  errors.value.username = null
 }
}

const validatePassword = (blur) => {
```

```javascript
  const password = formData.value.password
  const minLength = 8
  const hasUppercase = /[A-Z]/.test(password)
  const hasLowercase = /[a-z]/.test(password)
  const hasNumber = /\d/.test(password)
  const hasSpecialChar = /[!@#$%^&*(),.?":{}|<>]/.test(password)


  if (password.length < minLength) {
   if (blur) errors.value.password = `Password must be at least ${minLength} characters
long.`
  } else if (!hasUppercase) {
   if (blur) errors.value.password = 'Password must contain at least one uppercase letter.'
  } else if (!hasLowercase) {
   if (blur) errors.value.password = 'Password must contain at least one lowercase letter.'
  } else if (!hasNumber) {
   if (blur) errors.value.password = 'Password must contain at least one number.'
  } else if (!hasSpecialChar) {
   if (blur) errors.value.password = 'Password must contain at least one special
character.'
  } else {
   errors.value.password = null
  }
}

const validateConfirmPassword = (blur) => {
 if (formData.value.password !== formData.value.confirmPassword) {
   if (blur) errors.value.confirmPassword = 'Passwords do not match.'
 } else {
   errors.value.confirmPassword = null
```

```
  }
}

const validateReasonFriend = (blur) => {
 if (formData.value.reason.includes('friend')) {
  if (blur) errors.value.reason = 'Great to have a friend'
 } else {
  errors.value.reason = null
 }
}
</script>

<template>
 <!-- 🗂 W3. Library Registration Form -->
 <div class="container mt-5">
  <div class="row">
   <div class="col-md-8 offset-md-2">
    <h1 class="text-center">🗂 W4. Library Registration Form</h1>
    <p class="text-center">
     This form now includes validation. Registered users are displayed in a data table
below
     (PrimeVue).
    </p>
    <form @submit.prevent="submitForm">
     <div class="row mb-3">
      <div class="col-md-6 col-sm-6">
       <label for="username" class="form-label">Username</label>
       <input
```

```html
        type="text"

        class="form-control"

        id="username"

        @blur="() => validateName(true)"

        @input="() => validateName(false)"

        v-model="formData.username"

      />

      <div v-if="errors.username" class="text-danger">{{ errors.username }}</div>

    </div>


    <div class="col-md-6 col-sm-6">

      <label for="gender" class="form-label">Gender</label>

      <select class="form-select" id="gender" v-model="formData.gender" required>

        <option value="male">Male</option>

        <option value="female">Female</option>

        <option value="other">Other</option>

      </select>

    </div>

  </div>


  <div class="row mb-3">

    <div class="col-md-6 col-sm-6">

      <label for="password" class="form-label">Password</label>

      <input

        type="password"

        class="form-control"

        id="password"

        @blur="() => validatePassword(true)"
```

```
          @input="() => validatePassword(false)"

          v-model="formData.password"

        />

        <div v-if="errors.password" class="text-danger">{{ errors.password }}</div>

      </div>


      <div class="col-md-6 col-sm-6">

      <label for="confirm-password" class="form-label">Confirm password</label>

        <input

          type="password"

          class="form-control"

          id="confirm-password"

          v-model="formData.confirmPassword"

          @blur="() => validateConfirmPassword(true)"

        />

        <div v-if="errors.confirmPassword" class="text-
danger">{{ errors.confirmPassword }}</div>

        </div>


      </div>


      <div class="mb3">

        <div class="form-check">

        <input

          type="checkbox"

          class="form-check-input"

          id="isAustralian"

          v-model="formData.isAustralian"
```

```
          />
          <label class="form-check-label" for="isAustralian">Australian
Resident?</label>
        </div>
      </div>


      <div class="mb-3">
        <label for="reason" class="form-label">Reason for joining</label>
        <textarea
          class="form-control"
          id="reason"
          rows="3"
          v-model="formData.reason"
          @blur="() => validateReasonFriend(true)"
          @input="() => validateReasonFriend(false)"
        ></textarea>
      </div>
      <div v-if="errors.reason" class="text-danger">{{ errors.reason }}</div>
      <div class="text-center">
        <button type="submit" class="btn btn-primary me-2">Submit</button>
        <button type="button" class="btn btn-secondary"
@click="clearForm">Clear</button>
      </div>


      <div class="col-md-6 col-sm-6">
        <label for="suburb" class="form-label">Suburb</label>
          <input
            type="suburb"
            class="form-control"
```

```
            id="suburb"

            v-model="formData.suburb"

          />

          <!-- <div v-if="errors.suburb" class="text-danger">{{ errors.suburb }}</div> -->

        </div>

      </form>

    </div>

  </div>

</div>


<div class="row mt-5">

  <h4>This is a Primevue Datatable.</h4>

  <DataTable :value="submittedCards" tableStyle="min-width: 50rem">

    <Column field="username" header="Username"></Column>

    <Column field="password" header="Password"></Column>

    <Column field="isAustralian" header="Australian Resident"></Column>

    <Column field="gender" header="Gender"></Column>

    <Column field="reason" header="Reason"></Column>

  </DataTable>

</div>


<div class="row mt-5" v-if="submittedCards.length">

  <div class="d-flex flex-wrap justify-content-start">

    <div

      v-for="(card, index) in submittedCards"

      :key="index"

      class="card m-2"

      style="width: 18rem"
```

```
    >
      <div class="card-header">User Information</div>
      <ul class="list-group list-group-flush">
        <li class="list-group-item">Username: {{ card.username }}</li>
        <li class="list-group-item">Password: {{ card.password }}</li>
        <li class="list-group-item">
          Australian Resident: {{ card.isAustralian ? 'Yes' : 'No' }}
        </li>
        <li class="list-group-item">Gender: {{ card.gender }}</li>
        <li class="list-group-item">Reason: {{ card.reason }}</li>
      </ul>
    </div>
  </div>
</div>
</template>


<style scoped>
.container {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  max-width: 80vw;
  margin: 0 auto;
  padding: 20px;
  /* background-color: #e0bfbf; */
  border-radius: 10px;
}

/* Class selectors */
.form {
```

```css
  text-align: center;

  margin-top: 50px;

}


/* ID selectors */

#username:focus,

#password:focus,

#isAustralian:focus,

.card {

  border: 1px solid #ccc;

  border-radius: 10px;

  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.card-header {

  background-color: #275fda;

  color: white;

  padding: 10px;

  border-radius: 10px 10px 0 0;

}

.list-group-item {

  padding: 10px;

}
```
</style>