# FIT5032 Efolio 5

By Sathwik P. Kallapur, 34556834

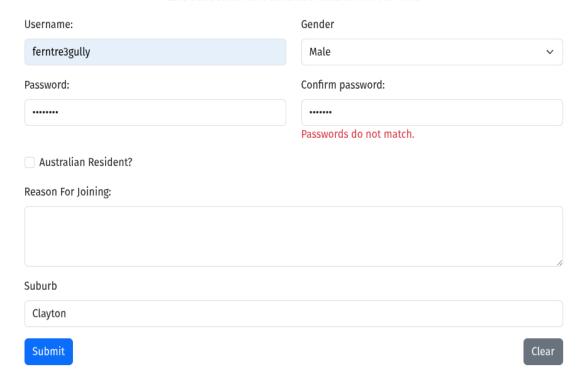
## Efolio task 5.1:

#### Screenshot 1:



# **W5. Library Registration Form**

Let's build some more advanced features into our form.

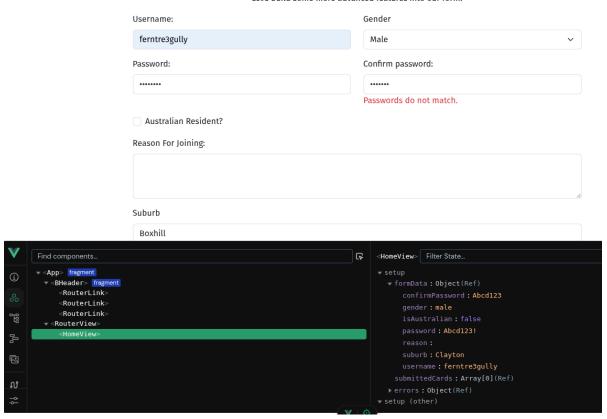


#### Screenshot 2:



# **W5. Library Registration Form**

Let's build some more advanced features into our form.



On the left hand side of the vue devTools panel, we can see that password !== confirmPassword, hence we see the error message in the form.

Also, despite entering "Boxhill" in Suburb, our formData.suburb remains "Clayton" because we used v-bind instead of v-model.

## Efolio task 5.2:

## router/index.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 AccessDenied from '../views/AccessDenied.vue'
const routes = [
 { path: '/', name: 'Home', component: HomeView },
 { path: '/login', name: 'Login', component: LoginView },
 { path: '/access-denied', name: 'AccessDenied', component:
AccessDenied },
  path: '/about',
  component: AboutView,
  meta: { requiresAuth: true },
history: createWebHistory(),
router.beforeEach((to, from, next) => {
const isAuthenticated = localStorage.getItem('auth') === 'true'
if (to.meta.requiresAuth && !isAuthenticated) {
   next()
export default router
```

### src/App.vue

```
<script setup>
import BHeader from './components/BHeader.vue';
  line-height: 1.5;
  text-align: center;
@media (min-width: 1024px) {
    display: flex;
    padding-right: calc(var(--section-gap) / 2);
   .logo {
    margin: 0 2rem 0 0;
```

```
border: 1px solid #ccc;
border-radius: 10px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
background-color: #275FDA;
color: white;
padding: 10px;
border-radius: 10px 10px 0 0;
padding: 10px;
  display: flex;
  justify-content: space-between;
<script setup>
import BHeader from './components/BHeader.vue';
```

```
<header>
line-height: 1.5;
 text-align: center;
@media (min-width: 1024px) {
  display: flex;
   place-items: center;
  padding-right: calc(var(--section-gap) / 2);
 .logo {
   margin: 0 2rem 0 0;
 font-size: 2.5rem;
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;
}

/* @media (min-width: 576px) {
    .card {
        width: 100%;
    }

/* #buttons {
    display: flex;
    justify-content: space-between;
}
</style>
```

#### src/views/HomeView.vue

```
<div class="row">
       <h1>W5. Library Registration Form</h1>
        <form @submit.prevent="submitForm">
           <div class="row mb-3">
             <div class="col-sm-6">
               <label for="username"</pre>
class="form-label">Username:</label>
               <input type="text" class="form-control" id="username"</pre>
name="username" v-model="formData.username"
               @blur="validateUsername(true)"
@input="validateUsername(false)">
               <div v-if="errors.username" class="text-danger">{{
errors.username }}</div>
             <div class="col-sm-6">
               <label for="gender"</pre>
class="form-label">Gender</label><br>
               <select class="form-select" id="gender"</pre>
v-model="formData.gender" required>
                   <option value="female">Female</option>
                   <option value="other">Other</option>
           <div class="row">
```

```
class="form-label">Password:</label>
               <input type="password" class="form-control"</pre>
id="password" name="password" v-model="formData.password"
               @blur="validatePassword(true)"
@input="validatePassword(false)">
               <div v-if="errors.password" class="text-danger">{{
errors.password }}</div>
             <div class="col-sm-6 mb-3">
               <label for="confirm-password" class="form-label">Confirm
password:</label>
                   type="password"
                   class="form-control"
                   id="confirm-password"
                   name="confirm-password"
                   v-model="formData.confirmPassword"
                   @blur="validateConfirmPassword(true)"
@input="validateConfirmPassword(false)"
               <div v-if="errors.confirmPassword"</pre>
class="text-danger">{{ errors.confirmPassword }}</div>
           <div class="row mb-3">
             <div class="col-sm-6">
               <div class="form-check">
                 <input type="checkbox" class="form-check-input"</pre>
id="isAustralian" v-model="formData.isAustralian">
                 <label for="isAustralian"</pre>
class="form-check-label">Australian Resident?</label>
```

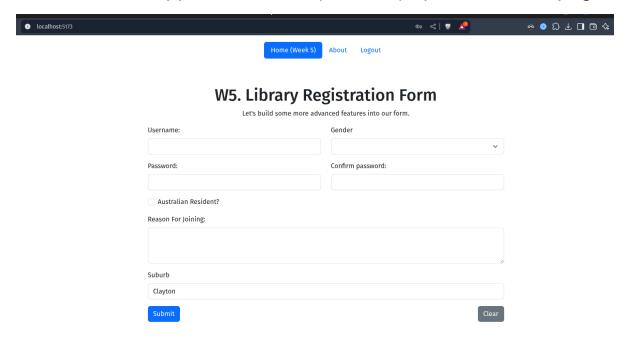
```
<div class="mb-3">
             <label for="reason" class="form-label">Reason For
Joining:</label>
             <textarea id="reason" rows="3" class="form-control"</pre>
v-model="formData.reason"></textarea>
           <div v-if="reasonHasFriend()" class="div mb-3">
             <div class="text" style="color: green;">it is great to
have a friend.</div>
           <div class="mb-3">
             <label for="reason" class="form-label">Suburb</label>
             <input type="text" class="form-control" id="suburb"</pre>
v-bind:value="formData.suburb" />
           <div id="buttons" >
             <div class="text-center">
               <button type="submit" class="btn</pre>
btn-primary">Submit</button>
             <div class="text-center">
@click.prevent="clearForm">Clear</button>
           <div class="row mt-5" v-if="submittedCards.length">
            <div class="col-12">
               <DataTable :value="submittedCards"</pre>
tableStyle="min-width: 50rem">
                 <Column field="username" header="Username"></Column>
                 <Column field="password" header="Password"></Column>
                 <Column field="isAustralian" header="Australian</pre>
Resident"></Column>
                 <Column field="gender" header="Gender"></Column>
                 <Column field="reason" header="Reason"></Column>
```

```
</DataTable>
<script setup>
   username: '',
   password: '',
   confirmPassword: '',
   isAustralian: false,
   gender: '',
   suburb: 'Clayton'
});
   password: '',
   confirmPassword: '',
   reason: '',
   gender: '',
   suburb: ''
```

```
const errors = ref({
  username: null,
  password: null,
  confirmPassword: null,
  isAustralian: null,
  reason: null,
  gender: null,
const validateUsername = (blur) => {
  if (formData.value.username.length < 3) {</pre>
     if (blur) errors.value.username = 'Username must be at least 3
 const reasonHasFriend = () => {
  return formData.value.reason.toLowerCase().includes('friend');
 const validatePassword = (blur) => {
  const password = formData.value.password;
  const hasUppercase = /[A-Z]/.test(password);
  const hasLowercase = /[a-z]/.test(password);
  const hasNumber = /\d/.test(password);
  if (password.length < minLength) {</pre>
     if (blur) errors.value.password = `Password must be at least
${minLength} characters long.`;
     if (blur) errors.value.password = "Password must contain at least
one uppercase letter.";
     if (blur) errors.value.password = "Password must contain at least
one lowercase letter.";
```

```
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.";
    errors.value.password = null;
const validateConfirmPassword = (blur) => {
  if (formData.value.password !== formData.value.confirmPassword) {
    if (blur) errors.value.confirmPassword = 'Passwords do not match.'
    errors.value.confirmPassword = null
 const submitForm = () => {
  validateUsername(true);
  validatePassword(true);
  if (! ( errors.value.username || errors.value.password ) ) {
    errors.value.username = null;
    submittedCards.value.push({
      ...formData.value
  clearForm();
```

# Screenshot of App in Localhost (browser) open on Home page



# Screenshot of App in Localhost (browser) open on About page

