

# FIT5032 Efolio 5

By Sathwik P. Kallapur, 34556834

## Efolio task 5.1:

Screenshot 1:

[Home \(Week 5\)](#) [About](#) [Logout](#)

### W5. Library Registration Form

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

Username:	Gender
<input type="text" value="ferntre3gully"/>	<input type="text" value="Male"/>
Password:	Confirm password:
<input type="password" value="....."/>	<input type="password" value="....."/>

Passwords do not match.

☐ Australian Resident?

Reason For joining:

Suburb

## Screenshot 2:

[Home \(Week 5\)](#) [About](#) [Logout](#)

### W5. Library Registration Form

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

Username:

Gender

Password:

Confirm password:

☐ Australian Resident?

Reason For Joining:

Suburb

Passwords do not match.

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',
    name: 'About',
    component: AboutView,
    meta: { requiresAuth: true },
  },
]

const router = createRouter({
  history: createWebHistory(),
  routes,
})

router.beforeEach((to, from, next) => {
  const isAuthenticated = localStorage.getItem('auth') === 'true'
  if (to.meta.requiresAuth && !isAuthenticated) {
    next('/login')
  } else {
    next()
  }
})

export default router
```

## src/App.vue

```
<script setup>
// import JSON from './components/JSON.vue'
import BHeader from './components/BHeader.vue';
// import Form from './views/HomeView.vue';
</script>

<template>
  <header>
    <BHeader />
  </header>

  <main>
    <!-- <JSON /> -->
    <!-- <Form /> -->
    <div class="container mt-5">
      <router-view />
    </div>
  </main>
</template>

<style >

  header {
    line-height: 1.5;
  }

  .container h1,p {
    text-align: center;
  }

  @media (min-width: 1024px) {
    header {
      display: flex;
      place-items: center;
      padding-right: calc(var(--section-gap) / 2);
    }

    .logo {
      margin: 0 2rem 0 0;
    }
  }
</style>
```

```

}

.h1 {
  color: #df0000;
  font-size: 2.5rem;
}

.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;
}

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

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

<script setup>
// import JSON from './components/JSON.vue'
import BHeader from './components/BHeader.vue';
// import Form from './views/HomeView.vue';
</script>

<template>

```

```
<header>
  <BHeader />
</header>

<main>
  <!-- <JSON /> -->
  <!-- <Form /> -->
  <div class="container mt-5">
    <router-view />
  </div>
</main>
</template>

<style >

header {
  line-height: 1.5;
}

.container h1,p {
  text-align: center;
}

@media (min-width: 1024px) {
  header {
    display: flex;
    place-items: center;
    padding-right: calc(var(--section-gap) / 2);
  }

  .logo {
    margin: 0 2rem 0 0;
  }

}

.h1 {
  color: #df0000;
  font-size: 2.5rem;
}

.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;
}

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

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

## src/views/HomeView.vue

```
<template>
  <div class="row">
    <div class="col-md-8 offset-md-2">
      <h1>W5. Library Registration Form</h1>
      <p>Let's build some more advanced features into our form.</p>

      <div class="form">
        <form @submit.prevent="submitForm">

          <div class="row mb-3">

            <div class="col-sm-6">
              <label for="username"
class="form-label">Username:</label>
              <input type="text" class="form-control" id="username"
name="username" v-model="formData.username"
@blur="validateUsername(true)"
@input="validateUsername(false)">
              <div v-if="errors.username" class="text-danger">{{
errors.username }}</div>
            </div>

            <div class="col-sm-6">
              <label for="gender"
class="form-label">Gender</label><br>
              <select class="form-select" id="gender"
v-model="formData.gender" required>
                <option value="female">Female</option>
                <option value="male">Male</option>
                <option value="other">Other</option>
              </select>
            </div>
          </div>
        </div>
      </div>

      <div class="row">

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



```

        <label for="password"
class="form-label">Password:</label>
        <input type="password" class="form-control"
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>

    <div class="col-sm-6 mb-3">
        <label for="confirm-password" class="form-label">Confirm
password:</label>

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

</div>

<div class="row mb-3">

    <div class="col-sm-6">
        <div class="form-check">
            <input type="checkbox" class="form-check-input"
id="isAustralian" v-model="formData.isAustralian">
            <label for="isAustralian"
class="form-check-label">Australian Resident?</label>
        </div>
    </div>

</div>

</div>

```

```

        <div class="mb-3">
            <label for="reason" class="form-label">Reason For
Joining:</label>
            <textarea id="reason" rows="3" class="form-control"
v-model="formData.reason"></textarea>
        </div>

        <div v-if="reasonHasFriend()" class="div mb-3">
            <div class="text" style="color: green;">it is great to
have a friend.</div>
        </div>

        <div class="mb-3">
            <label for="reason" class="form-label">Suburb</label>
            <input type="text" class="form-control" id="suburb"
v-bind:value="formData.suburb" />
        </div>

        <div id="buttons" >

            <div class="text-center">
                <button type="submit" class="btn
btn-primary">Submit</button>
            </div>

            <div class="text-center">
                <button class="btn btn-secondary"
@click.prevent="clearForm">Clear</button>
            </div>

        </div>

        <div class="row mt-5" v-if="submittedCards.length">
            <div class="col-12">

                <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>

    </form>
  </div>

</div>
</div>
</template>

<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 clearForm = () => {
  formData.value = {
    username: '',
    password: '',
    confirmPassword: '',
    isAustralian: false,
    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) {
    if (blur) errors.value.username = 'Username must be at least 3
characters long';
  } else {
    errors.value.username = null;
  }
};

const reasonHasFriend = () => {
  return formData.value.reason.toLowerCase().includes('friend');
};

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

    validateUsername(true);
    validatePassword(true);

    if (! ( errors.value.username || errors.value.password ) ) {
        errors.value.username = null;
        submittedCards.value.push({
            ...formData.value
        });
        clearForm();
    } else {
        return; // Prevent submission if there are validation errors
    }

};

</script>

```

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

The screenshot shows a web browser window with the address bar displaying 'localhost:5173'. The browser's navigation bar includes links for 'Home (Week 5)', 'About', and 'Logout'. The main content area features a registration form titled 'W5. Library Registration Form' with the subtitle 'Let's build some more advanced features into our form.'.

The form contains the following fields and controls:

- Username:** A text input field.
- Gender:** A dropdown menu.
- Password:** A text input field.
- Confirm password:** A text input field.
- Australian Resident?:** A checkbox.
- Reason For joining:** A large text area.
- Suburb:** A text input field containing the value 'Clayton'.
- Submit:** A blue button.
- Clear:** A grey button.

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

