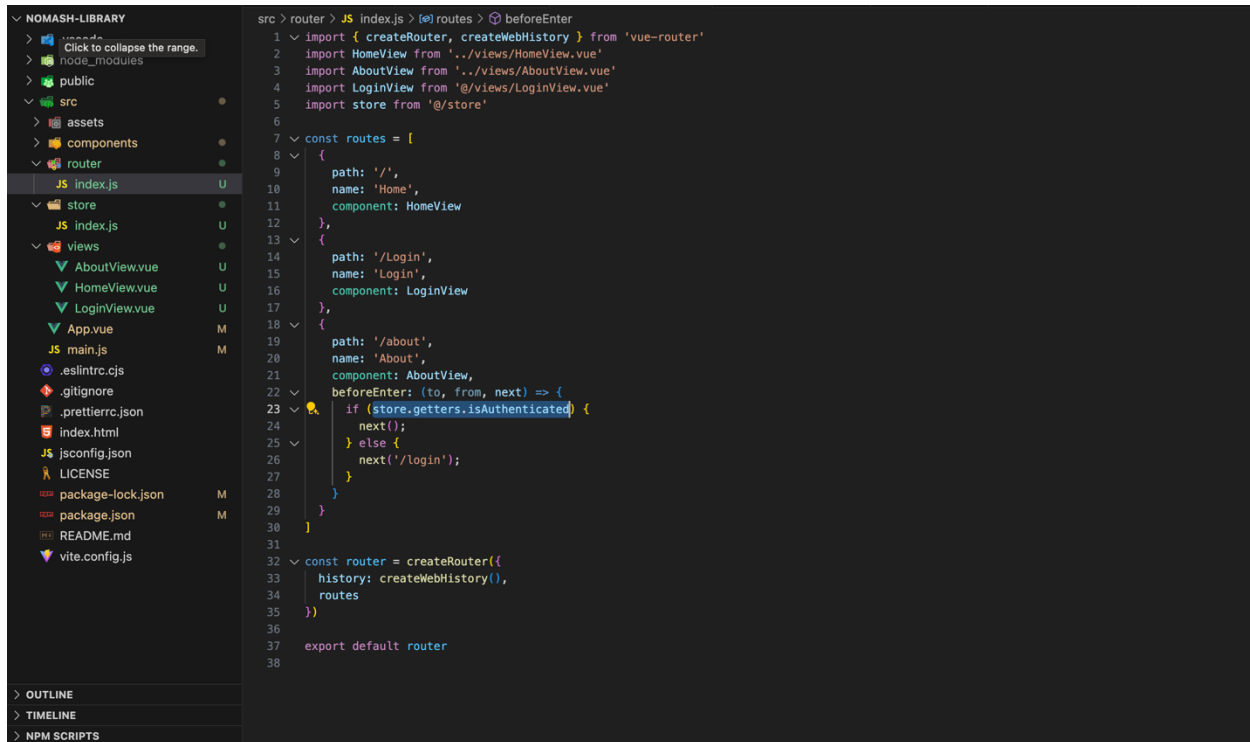
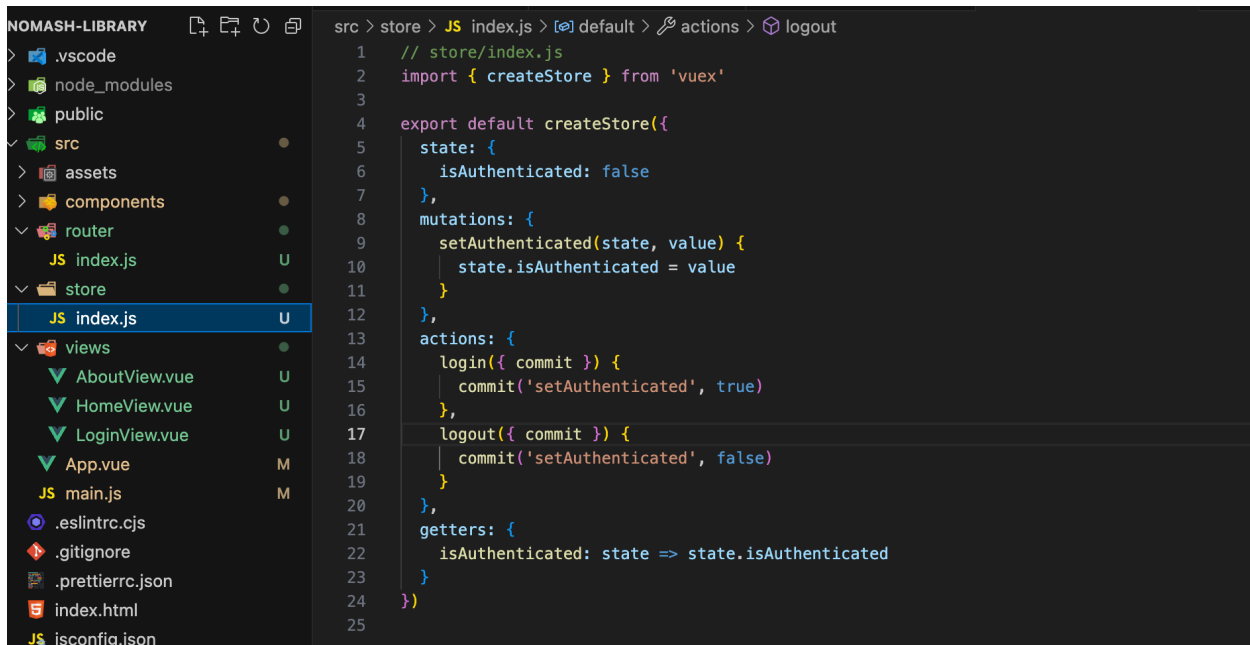


Router/index.js



```
src > router > JS index.js > beforeEnter
1 import { createRouter, createWebHistory } from 'vue-router'
2 import HomeView from '../views/HomeView.vue'
3 import AboutView from '../views/AboutView.vue'
4 import LoginView from '../views/LoginView.vue'
5 import store from '@store'
6
7 const routes = [
8   {
9     path: '/',
10    name: 'Home',
11    component: HomeView
12  },
13  {
14    path: '/Login',
15    name: 'Login',
16    component: LoginView
17  },
18  {
19    path: '/about',
20    name: 'About',
21    component: AboutView,
22    beforeEnter: (to, from, next) => {
23      if (store.getters.isAuthenticated) {
24        next()
25      } else {
26        next('/login')
27      }
28    }
29  }
30 ]
31
32 const router = createRouter({
33   history: createWebHistory(),
34   routes
35 })
36
37 export default router
38
```

Store/index.js (for storing the authentication status)



```
src > store > JS index.js > default > actions > logout
1 // store/index.js
2 import { createStore } from 'vuex'
3
4 export default createStore({
5   state: {
6     isAuthenticated: false
7   },
8   mutations: {
9     setAuthenticated(state, value) {
10       state.isAuthenticated = value
11     }
12 },
13 actions: {
14   login({ commit }) {
15     commit('setAuthenticated', true)
16   },
17   logout({ commit }) {
18     commit('setAuthenticated', false)
19   }
20 },
21 getters: {
22   isAuthenticated: state => state.isAuthenticated
23 }
24 })
25
```

HomeView

```
1  <script setup>
2  import { ref } from 'vue'
3  import DataTable from 'primevue/datatable'
4  import Column from 'primevue/column'
5
6  const formData = ref({
7    username: '',
8    password: '',
9    confirmPassword: '',
10   isAustralian: false,
11   reason: '',
12   gender: '',
13   suburb: 'Boxhill'
14 })
15
16 const submittedCards = ref([])
17
18 const submitForm = () => {
19   validateName(true)
20   validatePassword(true)
21   validateConfirmPassword(true)
22
23   if (!errors.value.username && !errors.value.password && !errors.value.confirmPassword) {
24     submittedCards.value.push({ ...formData.value })
25     clearForm()
26   }
27 }
28
29 const clearForm = () => {
30   formData.value = {
31     username: '',
32     password: '',
33     isAustralian: false,
34     reason: '',
35     gender: ''
36   }
37 }
38
39 const errors = ref({
40   username: null,
41   password: null,
42   confirmPassword: null,
43   resident: null,
44   gender: null,
45   reason: null
46 })
47
48 const validateName = (blur) => {
49   if (formData.value.username.length < 3) {
50     if (blur) errors.value.username = 'Name must be at least 3 characters'
51   } else {
52     errors.value.username = null
53   }
54 }
55
56 const validatePassword = (blur) => {
57   const password = formData.value.password
58   const minLength = 8
59   const hasUppercase = /[A-Z]/.test(password)
60   const hasLowercase = /[a-z]/.test(password)
61   const hasNumber = /\d/.test(password)
62   const hasSpecialChar = /[!@#$%^&*(),.?":{}|<>]/.test(password)
63
64   if (password.length < minLength) {
65     if (blur) errors.value.password = `Password must be at least ${minLength} characters long.`
66   } else if (!hasUppercase) {
67     if (blur) errors.value.password = 'Password must contain at least one uppercase letter.'
68   } else if (!hasLowercase) {
69     if (blur) errors.value.password = 'Password must contain at least one lowercase letter.'
70   } else if (!hasNumber) {
71     if (blur) errors.value.password = 'Password must contain at least one number.'
72   } else if (!hasSpecialChar) {
73     if (blur) errors.value.password = 'Password must contain at least one special character.'
74   } else {
75     errors.value.password = null
76   }
77 }
```

```

/**
 * Confirm password validation function that checks if the password and confirm password fields match.
 * @param blur: boolean - If true, the function will display an error message if the passwords do not match.
 */
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 checkFriend = (input) => {
  const lowerCaseReason = formData.value.reason.toLowerCase();
  if (lowerCaseReason.includes('friend')) {
    if (input) 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>

```

```

116       <input
117         type="text"
118         class="form-control"
119         id="username"
120         @blur="() => validateName(true)"
121         @input="() => validateName(false)"
122         v-model="formData.username"
123       />
124       <div v-if="errors.username" class="text-danger">{{ errors.username }}</div>
125     </div>
126
127     <div class="col-md-6 col-sm-6">
128       <label for="gender" class="form-label">Gender</label>
129       <select class="form-select" id="gender" v-model="formData.gender" required>
130         <option value="male">Male</option>
131         <option value="female">Female</option>
132         <option value="other">Other</option>
133       </select>
134     </div>
135
136
137   </div>
138   <div class="row mb-3">
139     <div class="col-md-6 col-sm-6">
140       <label for="password" class="form-label">Password</label>
141       <input
142         type="password"
143         class="form-control"
144         id="password"
145         @blur="() => validatePassword(true)"
146         @input="() => validatePassword(false)"
147         v-model="formData.password"
148       />
149       <div v-if="errors.password" class="text-danger">{{ errors.password }}</div>
150     </div>

```

```

151     <div class="col-md-6 col-sm-6">
152       <label for="confirm-password" class="form-label">Confirm password</label>
153       <input
154         type="password"
155         class="form-control"
156         id="confirm-password"
157         v-model="formData.confirmPassword"
158         @blur="() => validateConfirmPassword(true)"
159       />
160       <div v-if="errors.confirmPassword" class="text-danger">
161         {{ errors.confirmPassword }}
162       </div>
163     </div>
164
165   </div>
166   <div class="row mb-3">
167     <div class="col-md-6 col-sm-6">
168       <div class="form-check">
169         <input
170           type="checkbox"
171           class="form-check-input"
172           id="isAustralian"
173           v-model="formData.isAustralian"
174         />
175         <label class="form-check-label" for="isAustralian">Australian Resident?</label>
176       </div>
177     </div>
178   </div>
179   <div class="mb-3">
180     <label for="reason" class="form-label">Reason for joining</label>
181     <textarea
182       class="form-control"
183       id="reason"
184       rows="3"
185       v-model="formData.reason"
186
187       @input="() => checkFriend(true)"
188     ></textarea>
189     <div v-if="errors.reason" class="text-success">
190       {{ errors.reason }}
191     </div>
192   </div>
193   <div class="mb-3">
194     <label for="reason" class="form-label">Suburb</label>
195     <input type="text" class="form-control" id="suburb" v-bind:value="formData.suburb" />
196   </div>
197
198   <div class="text-center">
199     <button type="submit" class="btn btn-primary me-2">Submit</button>
200     <button type="button" class="btn btn-secondary" @click="clearForm">Clear</button>
201   </div>
202 </form>
203 </div>
204 </div>
205 </div>
206
207 <div class="row mt-5">
208   <h4>This is a Primevue Datatable.</h4>
209   <DataTable :value="submittedCards" tableStyle="min-width: 50rem">
210     <Column field="username" header="Username"></Column>
211     <Column field="password" header="Password"></Column>
212     <Column field="isAustralian" header="Australian Resident"></Column>
213     <Column field="gender" header="Gender"></Column>
214     <Column field="reason" header="Reason"></Column>
215   </DataTable>
216 </div>
217
218 <div class="row mt-5" v-if="submittedCards.length">
219   <div class="d-flex flex-wrap justify-content-start">
220     <div
221       v-for="(card, index) in submittedCards"
222       :key="index"

```

```

223     class="card m-2"
224     style="width: 18rem"
225   >
226     <div class="card-header">User Information</div>
227     <ul class="list-group list-group-flush">
228       <li class="list-group-item">Username: {{ card.username }}</li>
229       <li class="list-group-item">Password: {{ card.password }}</li>
230       <li class="list-group-item">
231         Australian Resident: {{ card.isAustralian ? 'Yes' : 'No' }}
232       </li>
233       <li class="list-group-item">Gender: {{ card.gender }}</li>
234       <li class="list-group-item">Reason: {{ card.reason }}</li>
235     </ul>
236   </div>
237 </div>
238 </div>
239 </template>
240
241 <style scoped>
242 .container {
243   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
244   max-width: 80vw;
245   margin: 0 auto;
246   padding: 20px;
247   /* background-color: #e0bfbf; */
248   border-radius: 10px;
249 }
250
251 /* Class selectors */
252 .form {
253   text-align: center;
254   margin-top: 50px;
255 }
256
257 /* ID selectors */
258 #username:focus,
259 #password:focus,
260 #isAustralian:focus,
261
262 .card {
263   border: 1px solid #ccc;
264   border-radius: 10px;
265   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
266 }
267 .card-header {
268   background-color: #275fda;
269   color: white;
270   padding: 10px;
271   border-radius: 10px 10px 0 0;
272 }
273 .list-group-item {
274   padding: 10px;
275 }
276 </style>

```

Login page

```
Open LoginView in Preview.js
1 <script setup>
2 import { ref } from 'vue'
3 import { useStore } from 'vuex'
4
5 const store = useStore()
6
7 const loginData = ref({
8   username: '',
9   password: ''
10 })
11
12 const account = ref({
13   username: "Andy",
14   password: "password"
15 })
16
17 const errors = ref({
18   username: null,
19   password: null
20 })
21
22 const validateUsername = (blur) => {
23   if (loginData.value.username !== account.value.username) {
24     if (blur) errors.value.username = 'Username does not exist'
25   } else {
26     errors.value.username = null
27   }
28 }
29
30 const validatePassword = (blur) => {
31   if (loginData.value.password !== account.value.password) {
32     if (blur) errors.value.password = 'Password is incorrect'
33   } else {
34     errors.value.password = null
35   }
36 }
37
38 const submitForm = () => {
39   validateUsername(true)
40   validatePassword(true)
41
42   if (!errors.value.username && !errors.value.password) {
43     store.dispatch('login')
44     alert("Login successful!")
45   }
46 }
47
48 const Logout = () => {
49   store.dispatch('logout')
50   alert("Logout successful!")
51 }
52 </script>
53
54
55 <template>
56   <div class="container mt-5">
57     <div class="row">
58       <div class="col-md-8 offset-md-2" v-if="!store.getters.isAuthenticated">
59         <h1 class="text-center">Log in</h1>
60         <p class="text-center">
61           This form is let user to login their account
62         </p>
63         <form @submit.prevent="submitForm">
64           <div class="row mb-3 d-flex justify-content-center">
65             <div class="col-md-6 col-sm-6">
66               <label for="username" class="form-label">Username</label>
67               <input
68                 type="text"
69                 class="form-control"
70                 id="username"
71                 @blur="() => validateUsername(true)"
72                 @input="() => validateUsername(false)"
73                 v-model="loginData.username"
74               />
75               <div v-if="errors.username" class="text-danger">{{ errors.username }}</div>
76             </div>
77
```

```

79     </div>
80     <div class="row mb-3 d-flex justify-content-center">
81       <div class="col-md-6 col-sm-6">
82         <label for="password" class="form-label">Password</label>
83         <input
84           type="password"
85           class="form-control"
86           id="password"
87           @blur="() => validatePassword(true)"
88           @input="() => validatePassword(false)"
89           v-model="loginData.password"
90         />
91         <div v-if="errors.password" class="text-danger">{{ errors.password }}</div>
92       </div>
93     </div>
94     <div class="text-center">
95       <button type="submit" class="btn btn-primary me-2">Submit</button>
96     </div>
97   </form>
98 </div>
99 <div class="col-md-8 offset-md-2" v-else>
100   <h1 class="text-center">You are already logged in</h1>
101   <p class="text-center">
102     You are already logged in. Please log out to log in with another account.
103   </p>
104   <div class="d-flex justify-content-center">
105     <button class="btn btn-primary me-2 d-flex justify-content-center" @click="Logout">Logout</button>
106   </div>
107 </div>
108 </div>
109 </div>
110 </template>
111
112
113 <style scoped>
114 .container {
115   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
116   max-width: 80vw;
117   margin: 0 auto;
118   padding: 20px;
119   border-radius: 10px;
120 }
121
122 /* ID selectors */
123 #username:focus,
124 #password:focus
125 {
126   border: 1px solid #ccc;
127   border-radius: 10px;
128   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
129 }
130
131 /* Class selectors */
132 .form {
133   text-align: center;
134   margin-top: 50px;
135 }
136 </style>
137

```

App.vue (below)

```
<script setup>

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

</script>

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

  <main>
    <router-view></router-view>
  </main>
</template>

<style scoped>
```

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



W4. Library Registration Form

This form now includes validation. Registered users are displayed in a data table below (PrimeVue).

Username

Gender

Password

Confirm password

☐ Australian Resident?

Reason for joining

Suburb

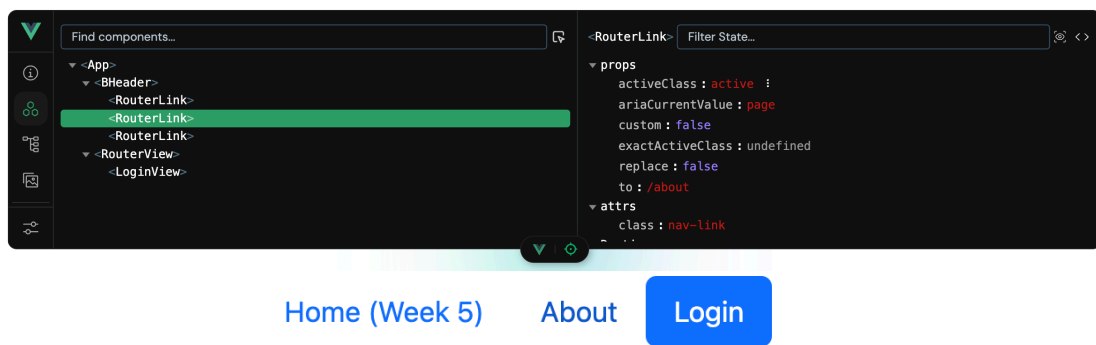
[Submit](#)

[Clear](#)

You are already logged in

You are already logged in. Please log out to log in with another account.

[Logout](#)



Log in

This form is let user to login their account

Username

Password

[Submit](#)

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

[About](#)

[Login](#)

About Our Library

Welcome to our digital library! We're dedicated to providing a vast collection of books and resources to our community.

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

[About](#)

[Login](#)

You are already logged in

You are already logged in. Please log out to log in with another account.

[Logout](#)

localhost:5174 says
Logout successful!
OK

You are already logged in

You are already logged in. Please log out to log in with another account.

Logout

Log in

This form is let user to login their account

Username

Andy

Password

.....

Password is incorrect

Submit