

EFOLIO TASK 3.1

User Information Form

☐ Australian Resident?

Gender

Female

Reason for joining

none


Submit


Clear


User Information	User Information
<div>Username: Yujie</div>	<div>Username: Mona</div>
<div>Password: 123</div>	<div>Password: 666</div>
<div>Australian Resident: No</div>	<div>Australian Resident: Yes</div>
<div>Gender: female</div>	<div>Gender: female</div>






EFOLIO TASK 3.2

GitHub commit history:

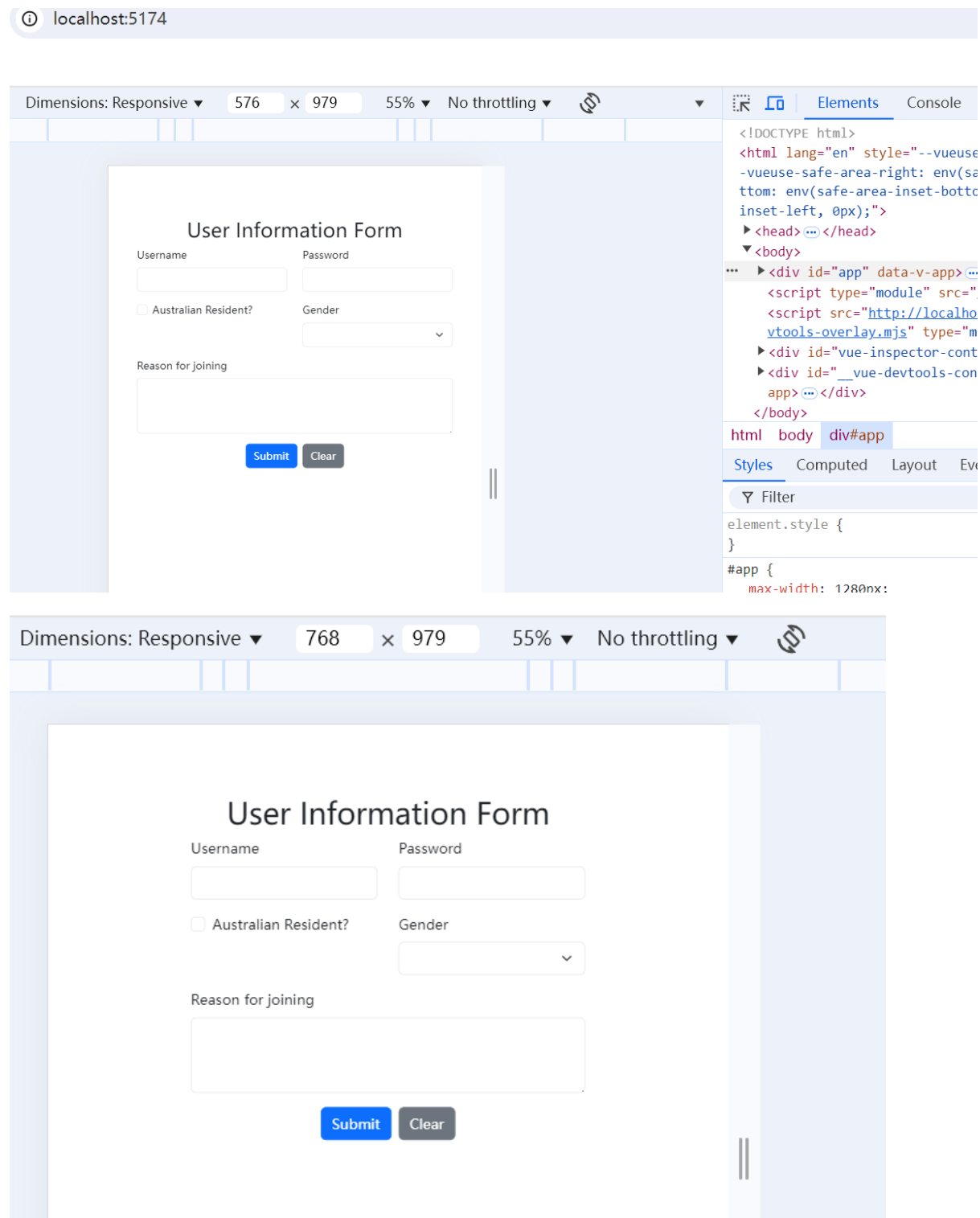
[https---github.com-yujie-huu-fit5032](https://github.com-yujie-huu-fit5032) / [src](#) / [components](#) / 


yujie-huu
efolio 3.2&4.1

f528a0c · 10 minutes ago
 History

Name	Last commit message	Last commit date
 ..		
 icons	efolio 1	2 weeks ago
 Donations.vue	efolio 1	2 weeks ago
 Form.vue	efolio 3.2&4.1	10 minutes ago
 HealthyEatingTips.vue	efolio 1	2 weeks ago

Screenshots of front-end demonstrating the form support on different devices:



Code for implementing the Bootstrap's breakpoints. (Edited codes are highlighted in blue):

```
1  <template>
2    <div class="container mt-5">
3      <div class="row">
4        <div class="col-md-8 offset-md-2">
5          <form @submit.prevent="submitForm">
6            <div class="col-sm-6">
7              <label for="username" class="form-label">Username</label>
8              <input
9                type="text"
10               class="form-control"
11               id="username"
12               @blur="() => validateName(true)"
13               @input="() => validateName(false)"
14               v-model="formData.username"
15             />
16             <div v-if="errors.username" class="text-danger">{{ errors.username }}</div>
17           </div>
18           <div class="col-sm-6">
19             <label for="password" class="form-label">Password</label>
20             <input
21               type="password"
22               class="form-control"
23               id="password"
24               @blur="() => validatePassword(true)"
25               @input="() => validatePassword(false)"
26               v-model="formData.password"
27             />
28             <div v-if="errors.password" class="text-danger">{{ errors.password }}</div>
29           </div>
30         </div>
31         <div class="row mb-3">
32           <div class="col-sm-6">
33             <div class="form-check">
34               <input
35                 type="checkbox"
36                 class="form-check-input"
37                 id="isAustralian"
38                 v-model="formData.isAustralian"
39               />
40               <label class="form-check-label" for="isAustralian">Australian Resident?</label>
41             </div>
42           </div>
43           <div class="col-sm-6">
44             <label for="gender" class="form-label">Gender</label>
45             <select
46               class="form-select"
47               id="gender"
48             />
49           </div>
50         </div>
51       </div>
52     </div>
53   </template>
```

Task 4.1

Activity 2:

User Information Form

Username Password

☐ Australian Resident?

Reason for joining

User Information Form

Username Password

yujie ..

☐ Australian Resident?

Reason for joining

User Information Form

Username Password

yujie

☐ Australian Resident? Gender

Reason for joining

User Information Form

Username Password

yujie

☐ Australian Resident? Gender Female

Reason for joining

Activity 3:

Empty input:

User Information Form

Username	Password
<input type="text"/>	<input type="password"/>
Name must be at least 3 characters	Password must be at least 8 characters long.
<input type="checkbox"/> Australian Resident?	Gender <input type="text"/>
	You have to select a gender from the list.
Reason for joining <input type="text"/>	
Reason must be at least 10 characters	
<input type="button" value="Submit"/>	<input type="button" value="Clear"/>

Incorrect inputs:

User Information Form

Username	Password
<input type="text" value="yujiehu"/>	<input type="password" value="....."/>
	Password must contain at least one uppercase letter.
<input type="checkbox"/> Australian Resident?	Gender <input type="text" value="Female"/>
Reason for joining <input type="text" value="12345"/>	
Reason must be at least 10 characters	
<input type="button" value="Submit"/>	<input type="button" value="Clear"/>

Correct Inputs:

User Information Form

Username	Password
<input type="text"/>	<input type="password"/>
<input type="checkbox"/> Australian Resident?	Gender <input type="text"/>
Reason for joining <input type="text"/>	
<input type="button" value="Submit"/>	<input type="button" value="Clear"/>

User Information	User Information
Username: yujiehu	Username: yujiehu
Password: e23!#F2bdf	Password: 12345Pass@
Australian Resident: No	Australian Resident: No
Gender: male	Gender: female
Reason: vefvvevswvqe	Reason: valid reason

Task 4.2

Extra validations of:

- no duplicate name
- must select a gender from the list
- minimum 10 characters length for join reason

User Information Form

Username

yujiehu

This username has already been submitted

Password

.....

☐ Australian Resident?

Gender

You have to select a gender from the list.

Reason for joining

Reason must be at least 10 characters

Submit

Clear

Username	Password	Australian Resident
yujiehu	e12e12e12\$6543D	false

PrimeVue DataTable with submitted user information:

User Information Form

Username

Password

☐ Australian Resident?

Gender

Reason for joining

Submit

Clear

Username	Password	Australian Resident
yujiehu	e12e12e12\$6543D	false
Mona	f23r12312\$F	false
yhuu0149	vwwe%\$DD12	false

Codes for extra validation:

```
9      <label for="username" class="form-label">Username</label>
10     <input
11       type="text"
12       class="form-control"
13       id="username"
14       @blur="() => validateName(true) && validateUser(true)"
15       @input="() => validateName(false) && validateUser(false)"
16       v-model="formData.username"
17     />
18     <div v-if="errors.username" class="text-danger">{{ errors.username }}</div>

46     <label for="gender" class="form-label">Gender</label>
47     <select
48       class="form-select"
49       id="gender"
50       @blur="() => validateGender(true)"
51       @input="() => validateGender(false)"
52       v-model="formData.gender"
53     >
54       <option value="male">Male</option>
55       <option value="female">Female</option>
56       <option value="other">Other</option>
57     </select>
58     <div v-if="errors.gender" class="text-danger">{{ errors.gender }}</div>

62     <label for="reason" class="form-label">Reason for joining</label>
63     <textarea
64       class="form-control"
65       id="reason"
66       rows="3"
67       @blur="() => validateReason(true)"
68       @input="() => validateReason(false)"
69       v-model="formData.reason"
70     ></textarea>
71     <div v-if="errors.reason" class="text-danger">{{ errors.reason }}</div>

108    const submitForm = () => {
109      validateName(true)
110      validatePassword(true)
111      validateGender(true)
112      validateReason(true)
113      validateUser(true)
114      if (
115        !errors.value.username &&
116        !errors.value.password &&
117        !errors.value.gender &&
118        !errors.value.reason
119      ) {
120        submittedCards.value.push({ ...formData.value })
121        clearForm()
122      }
123    }
```

```

174 const validateGender = (blur) => {
175   const gender = formData.value.gender
176   const selected = /^(female|male|other)$/.test(gender)
177
178   if (!selected) {
179     if (blur) errors.value.gender = 'You have to select a gender from the list.'
180   } else {
181     errors.value.gender = null
182   }
183 }
184
185 const validateReason = (blur) => {
186   if (formData.value.reason.length < 10) {
187     if (blur) errors.value.reason = 'Reason must be at least 10 characters'
188   } else {
189     errors.value.reason = null
190   }
191 }
192
193 const validateUser = (blur) => {
194   // Check if the username already exists in the submittedCards array
195   const isDuplicate = submittedCards.value.some((card) => card.username === formData.value.username)
196
197   if (isDuplicate) {
198     errors.value.username = 'This username has already been submitted'
199   } else {
200     errors.value.username = null
201   }
202 }

```


Codes for PrimeVue DataTable:



```



73 <div class="text-center">
74   <button type="submit" class="btn btn-primary me-2">Submit</button>
75   <button type="button" class="btn btn-secondary" @click="clearForm">Clear</button>
76 </div>
77 <div class="row mt-5" v-if="submittedCards.length">
78   <DataTable :value="submittedCards" tableStyle="min-width: 50rem">
79     <Column field="username" header="Username"></Column>
80     <Column field="password" header="Password"></Column>
81     <Column field="isAustralian" header="Australian Resident"></Column>
82     <Column field="gender" header="Gender"></Column>
83     <Column field="reason" header="Reason for joining"></Column>
84   </DataTable>
85 </div>

```

GitHub commit history:

<https---github.com-yujie-huu-fit5032> / src / 

Add file  

 yujie-huu efolio task 4.2 f49f4a6 · 1 minute ago  History

Name	Last commit message	Last commit date
..		
assets	efolio 1	2 weeks ago
components	efolio task 4.2	1 minute ago
App.vue	environment setup	last week
main.js	efolio task 4.2	1 minute ago
style.css	environment setup	last week