[eFolio activity5 & 6]

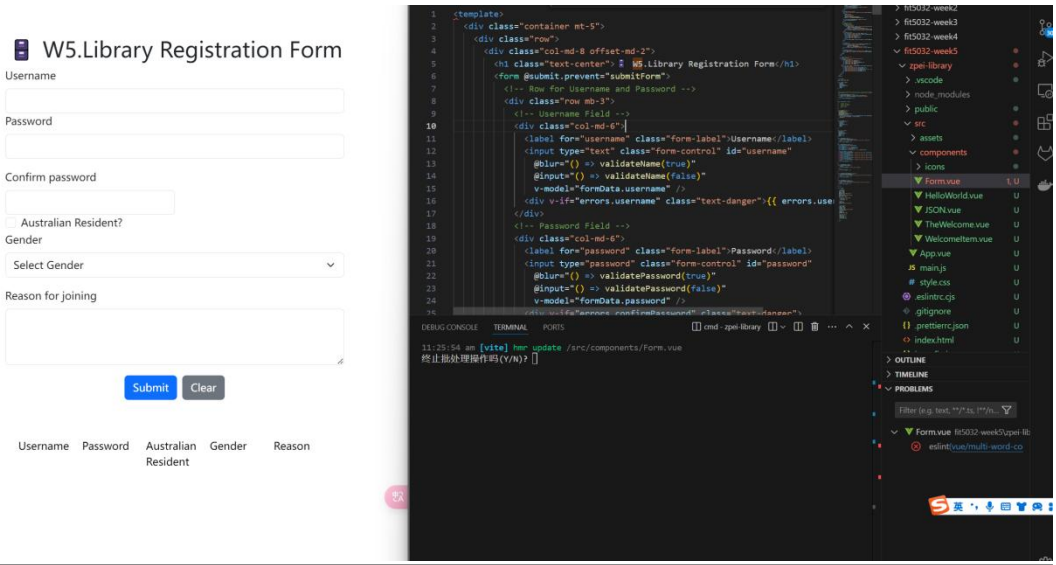
Ziqi Pei 33429472

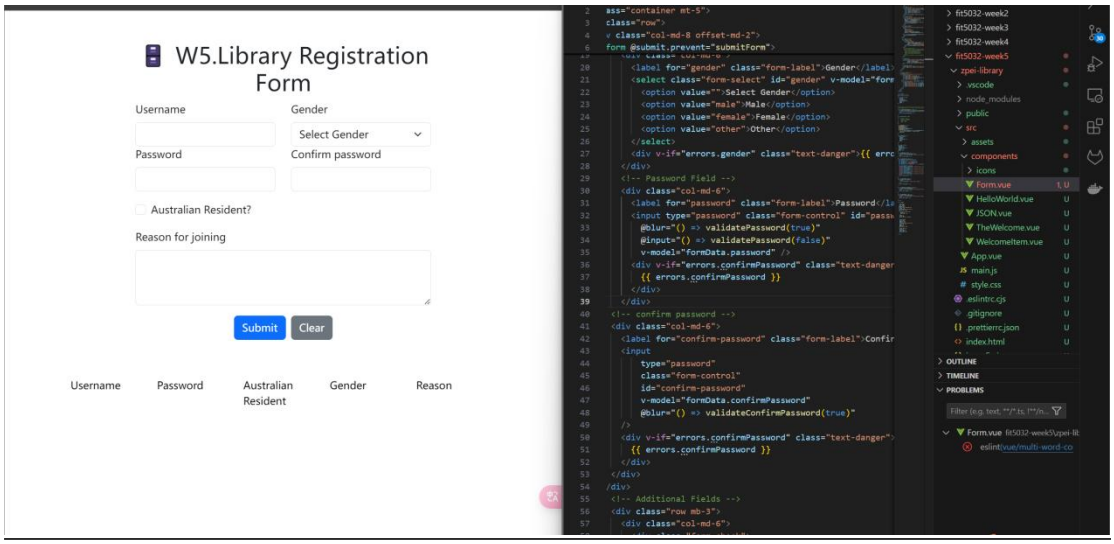
Lab Thusday 10am-12am

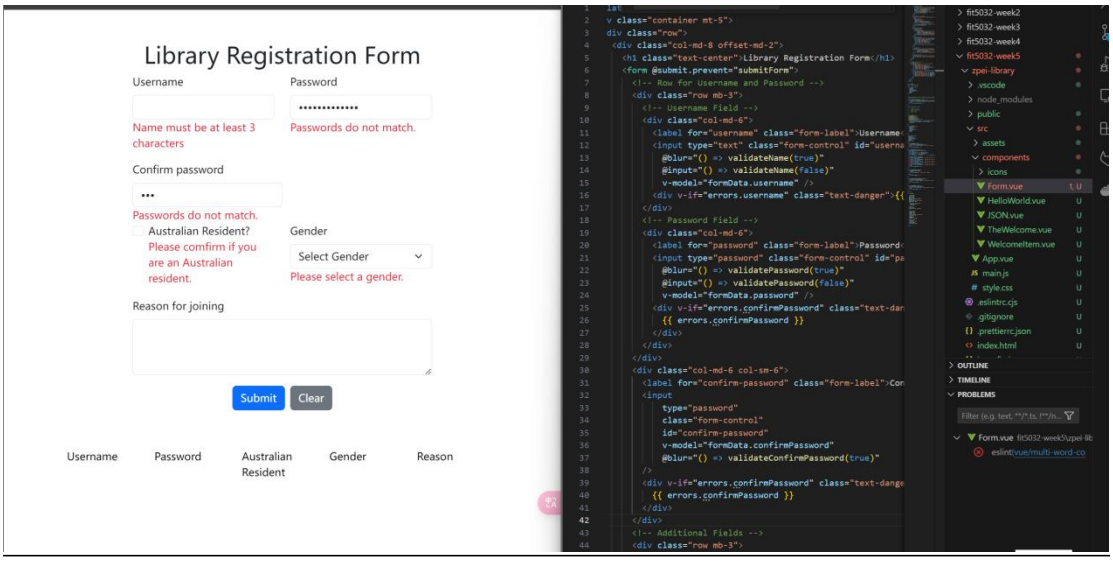
Github rep:

https://github.com/Franz-Pei/Franz/tree/881769a859b603ccc0f8cac91d9818b70a5c24e5/MONASH_IT_MASTER/FIT5032Internet_applications_development

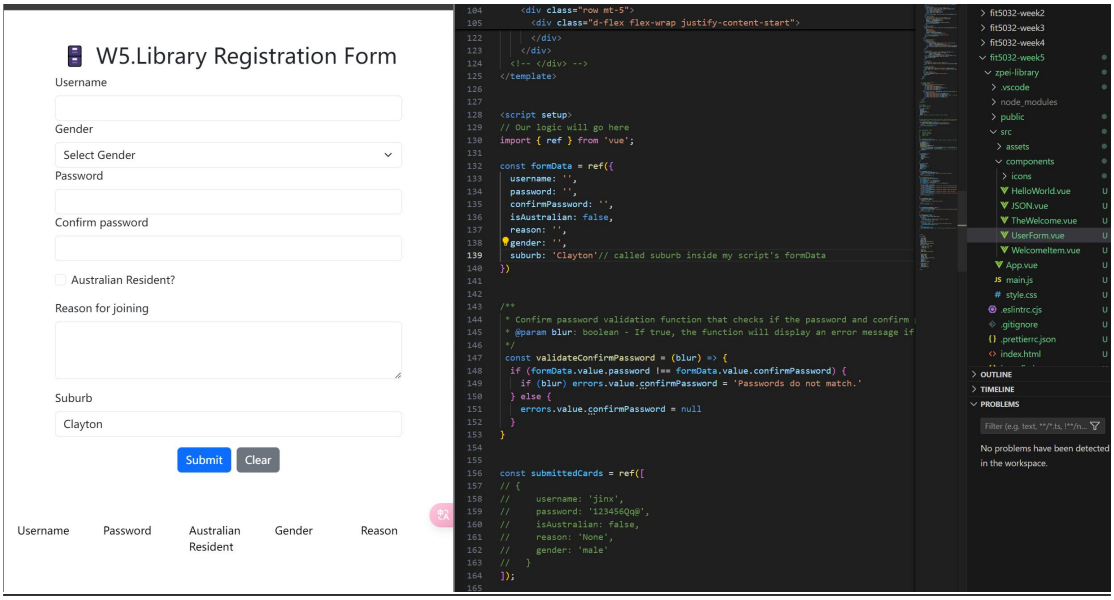
Activity 5.3.1: Setup the Form

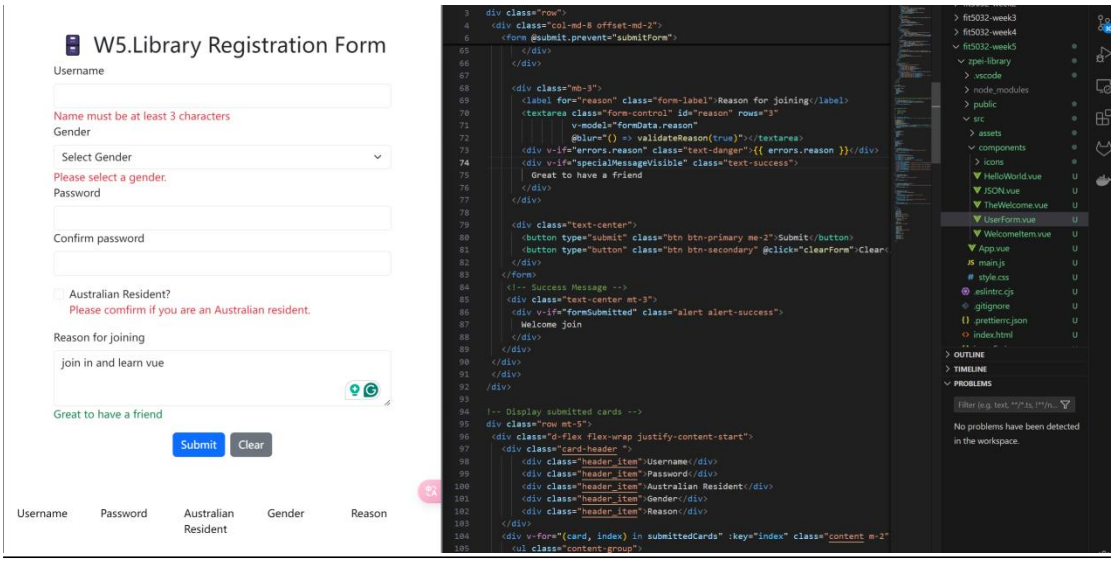




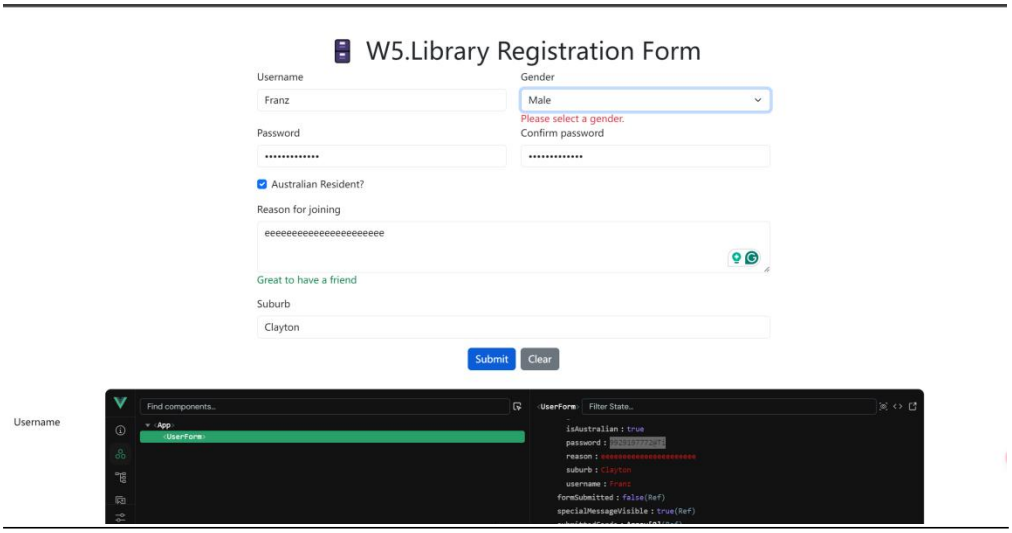


ctivity 5.3.2 Improve validation for "Reason for joining"

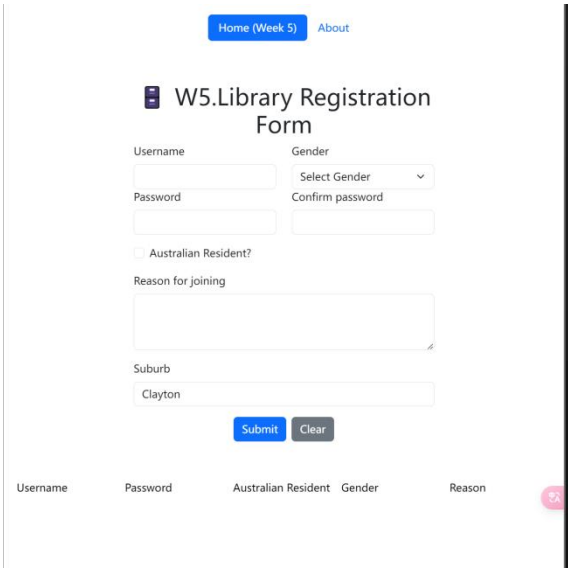


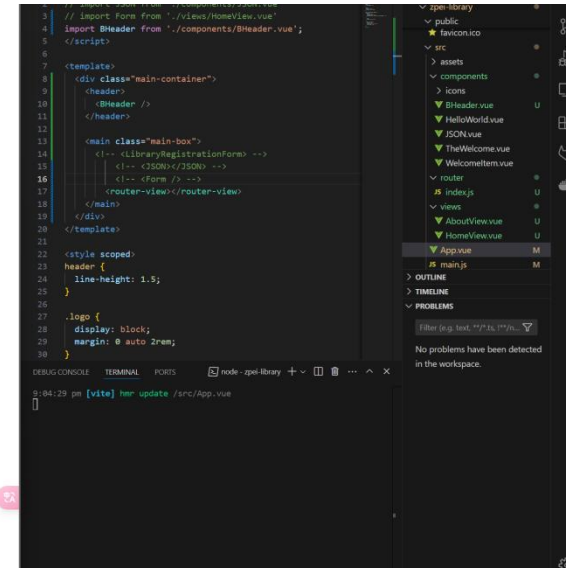


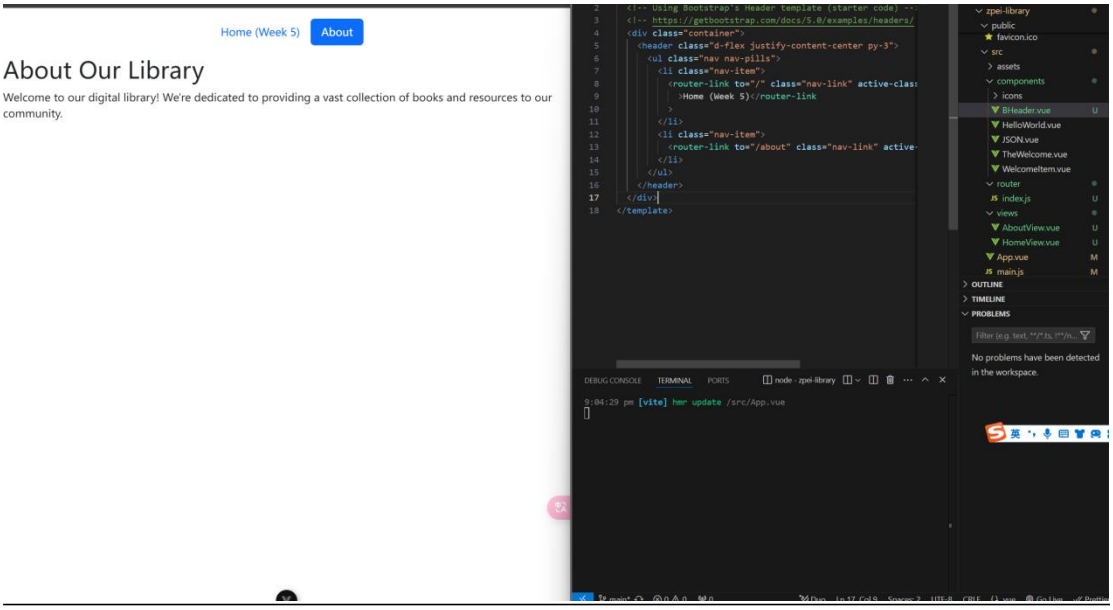
4. Benefit of using v-mode

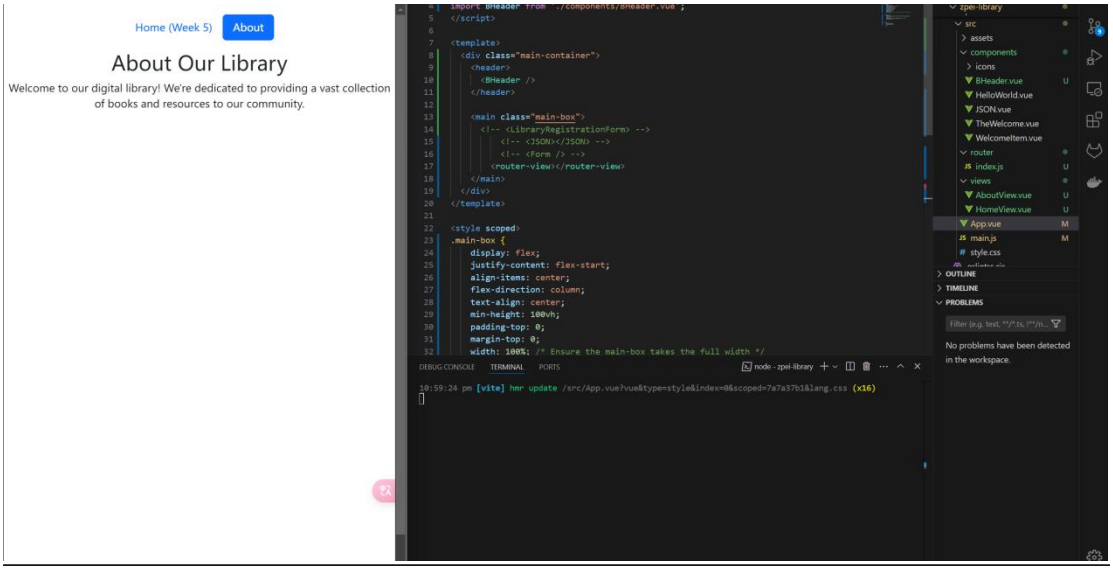


Activity 5.5.1 Installing and Setup of Router

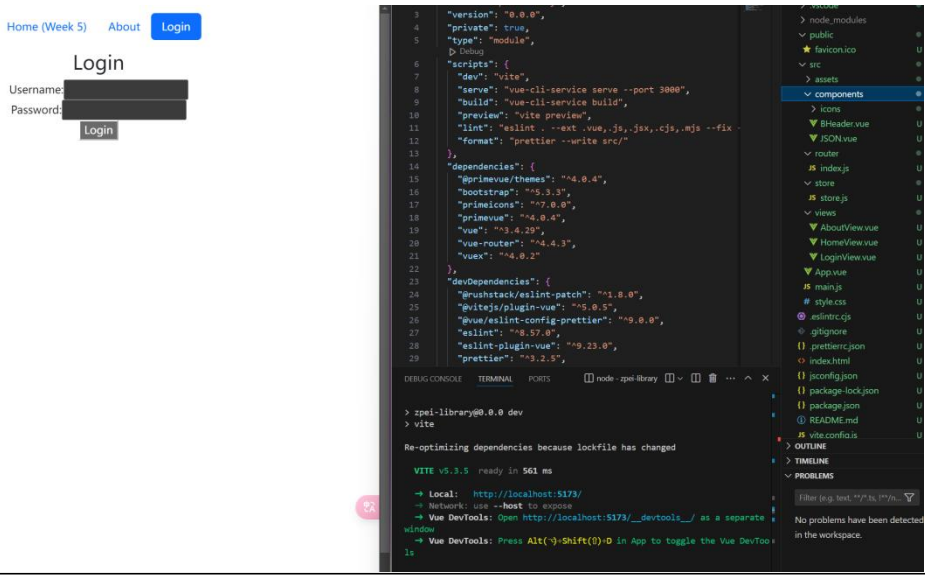


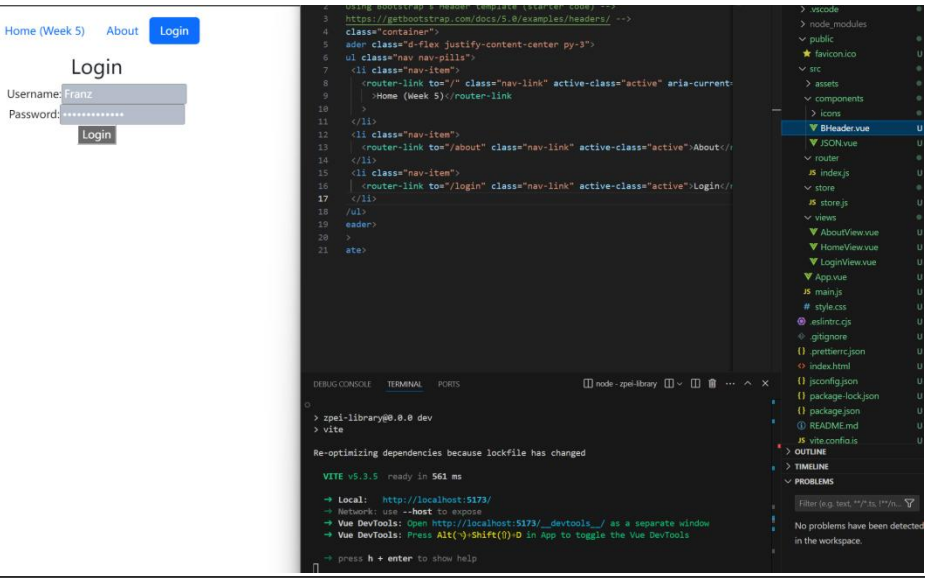




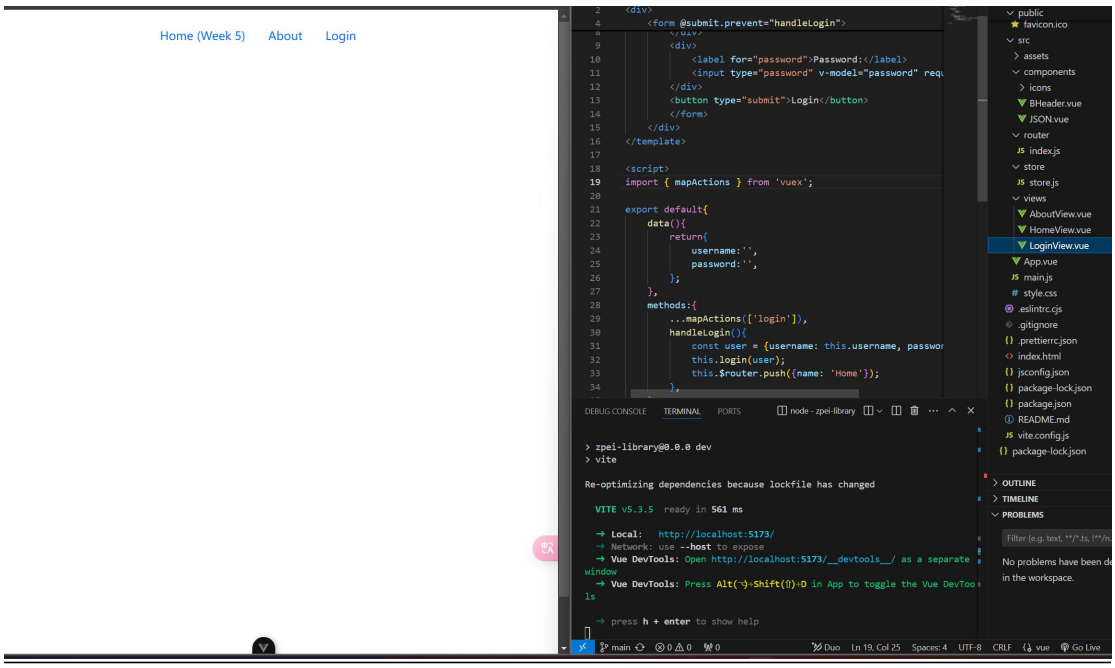


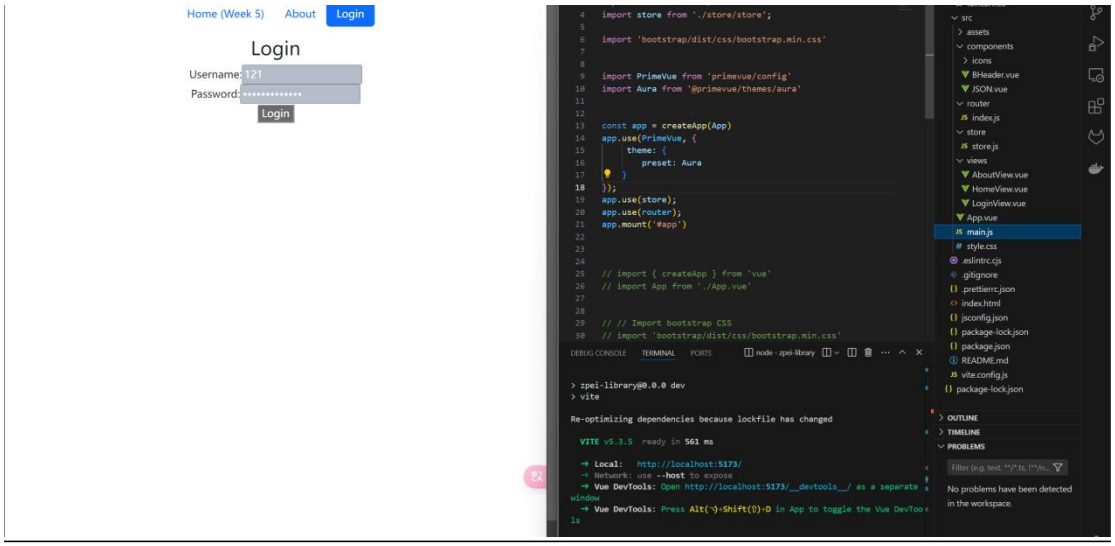
Activity 6.1: Creating a Secure Members Area in Your Library App





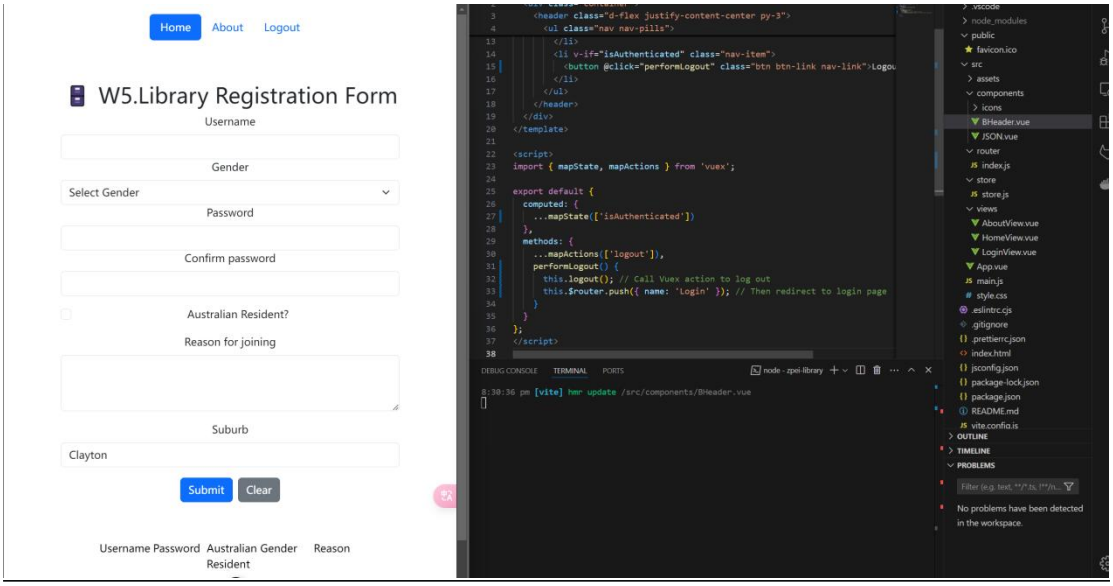
If user not login user can't access the homepage

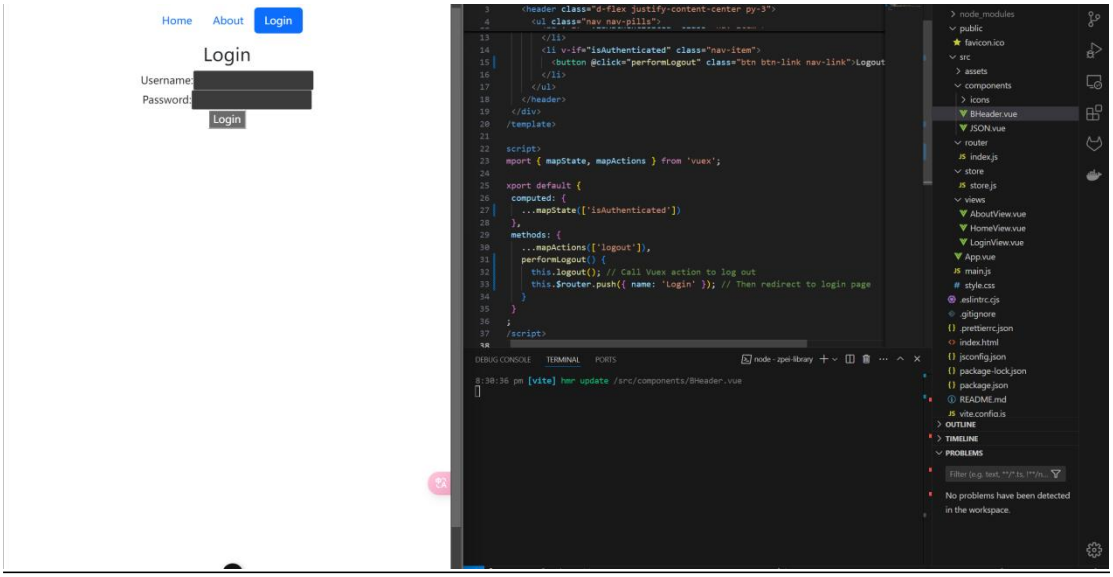




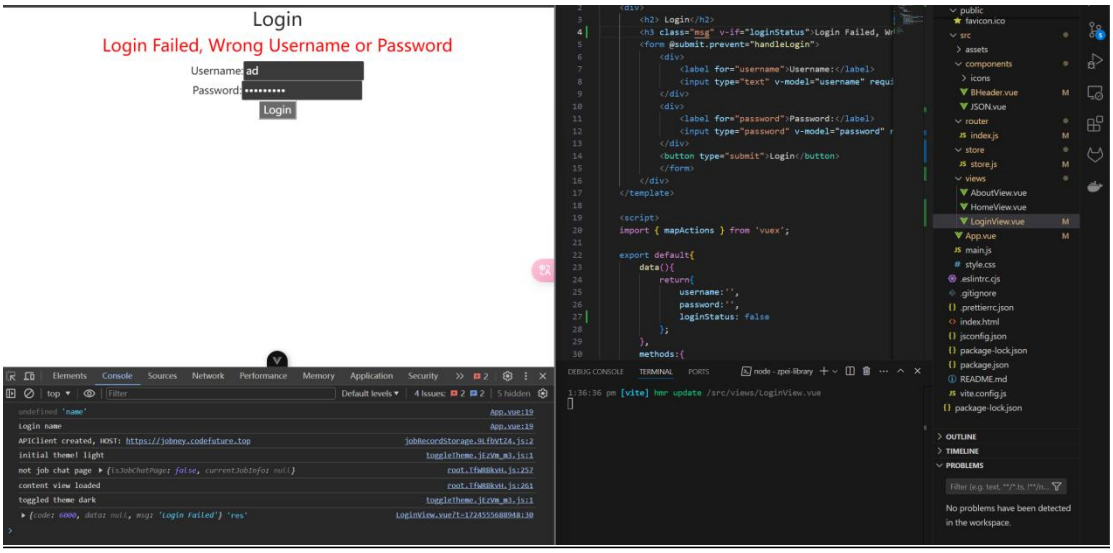
Conditional Rendering

Use `v-if` and `v-else` directives to show different content based on authentication status

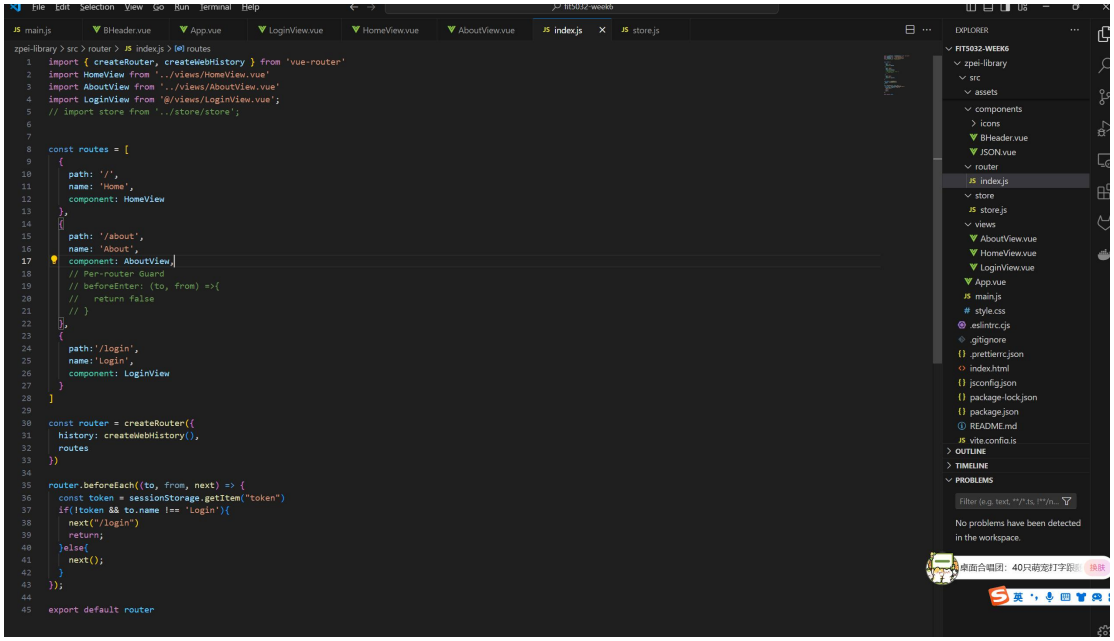




If user is wrong account not the static user set will jump a word

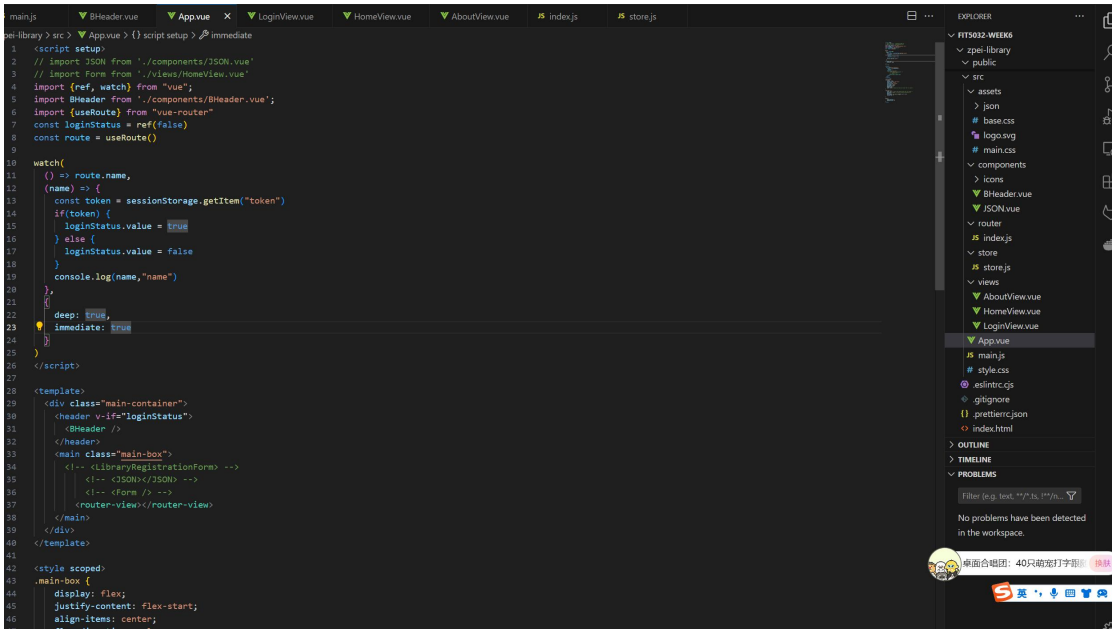


Router.index.js

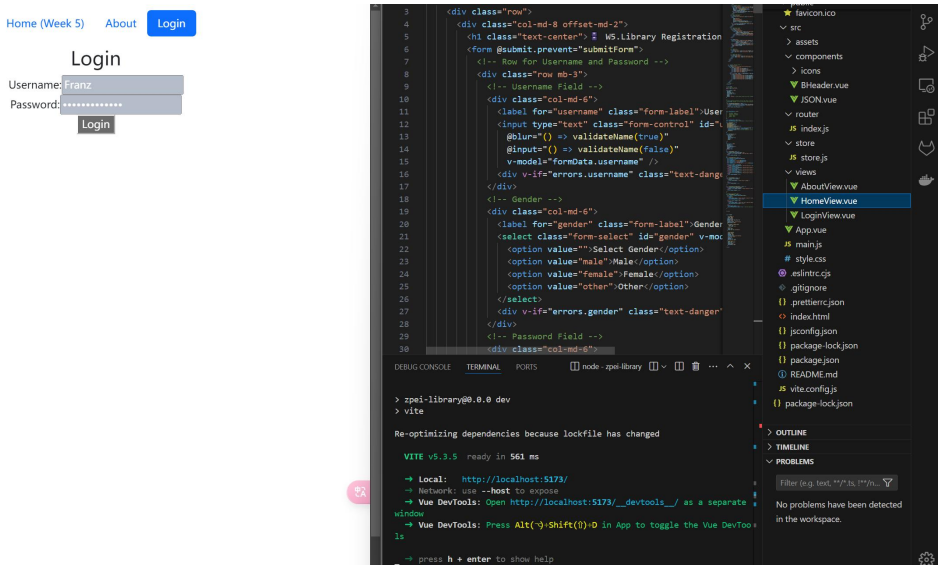


src/App.vue

If else

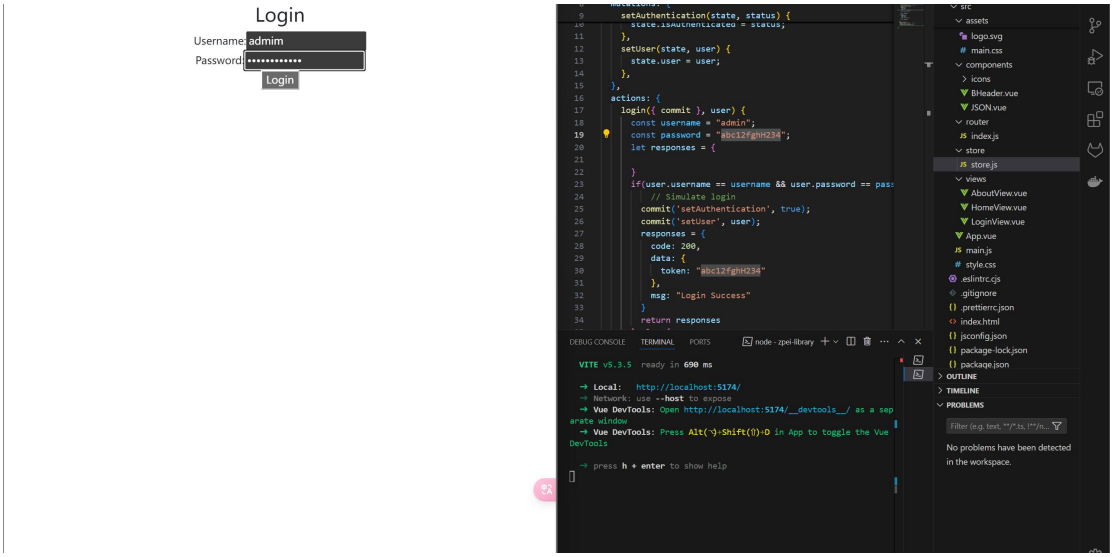


Screenshot of App in Localhost (browser) open on Home page

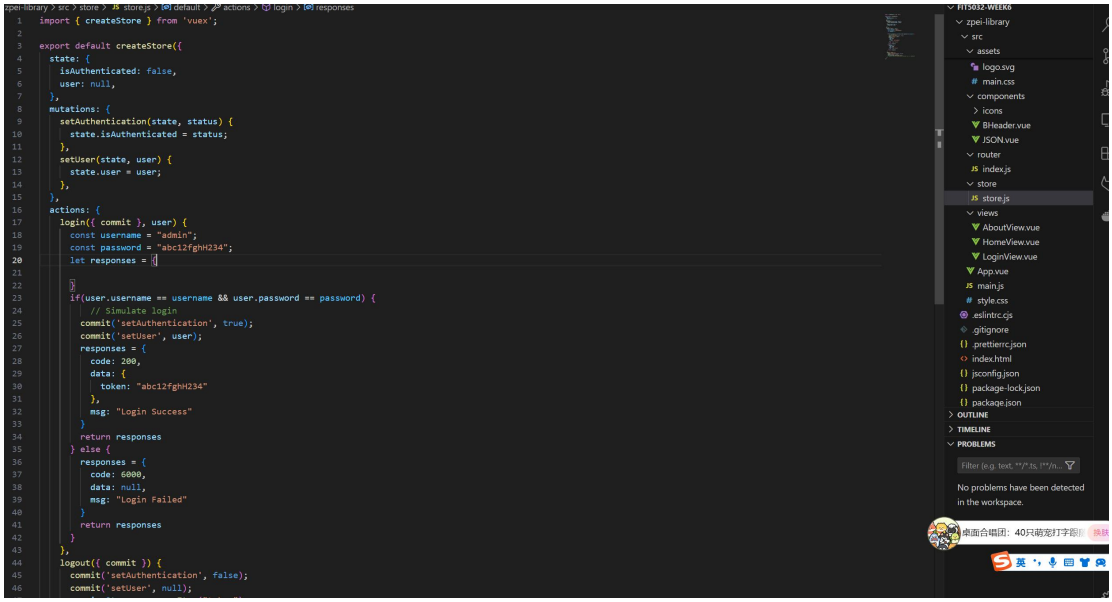


Screenshot of App in Localhost (browser) open on About page

Close

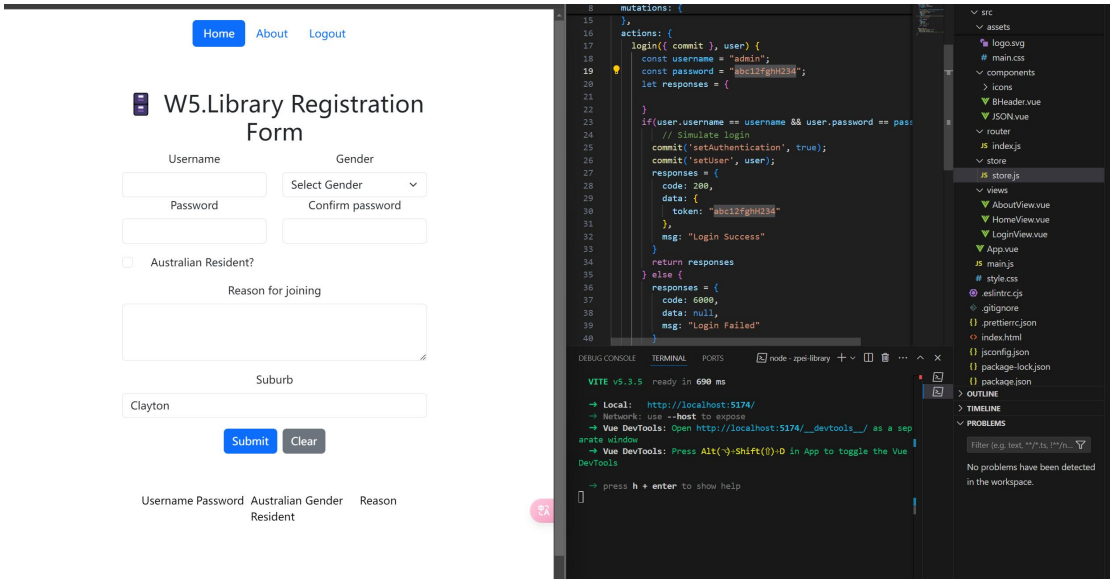


Store.js static data set

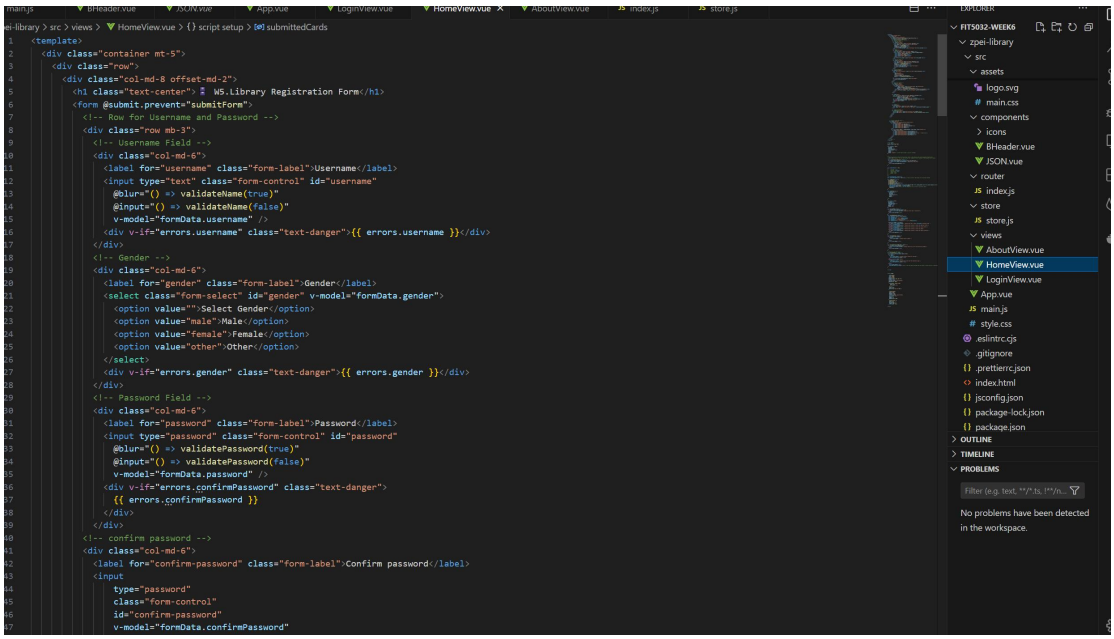


```
import { createStore } from 'vuex';

export default createStore({
  state: {
    isAuthenticated: false,
    user: null,
  },
  mutations: {
    setAuthentication(state, status) {
      state.isAuthenticated = status;
    },
    setUser(state, user) {
      state.user = user;
    },
  },
  actions: {
    login({ commit }, user) {
      const username = "admin";
      const password = "abc12fgHH234";
      let responses = {

      }
      if(user.username == username && user.password == password) {
          // Simulate login
        commit('setAuthentication', true);
        commit('setUser', user);
        responses = {
          code: 200,
          data: {
            token: "abc12fgHH234"
          },
          msg: "Login Success"
        }
        return responses
      } else {
        responses = {
          code: 6000,
          data: null,
          msg: "Login Failed"
        }
        return responses
      }
    },
    logout({ commit }) {
      commit('setAuthentication', false);
      commit('setUser', null);
```

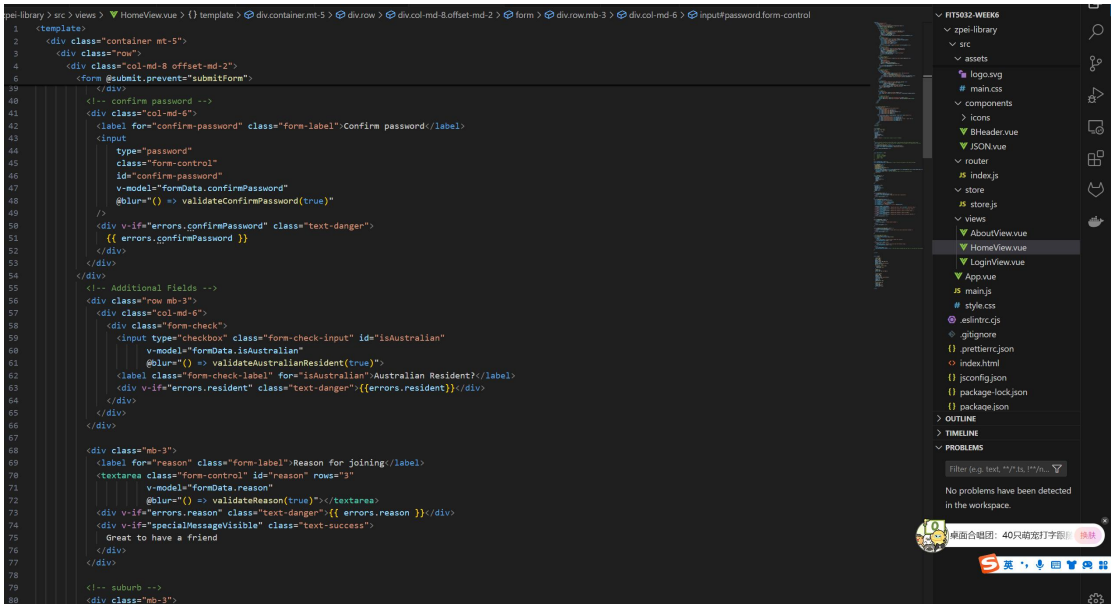
When login success logout show can click



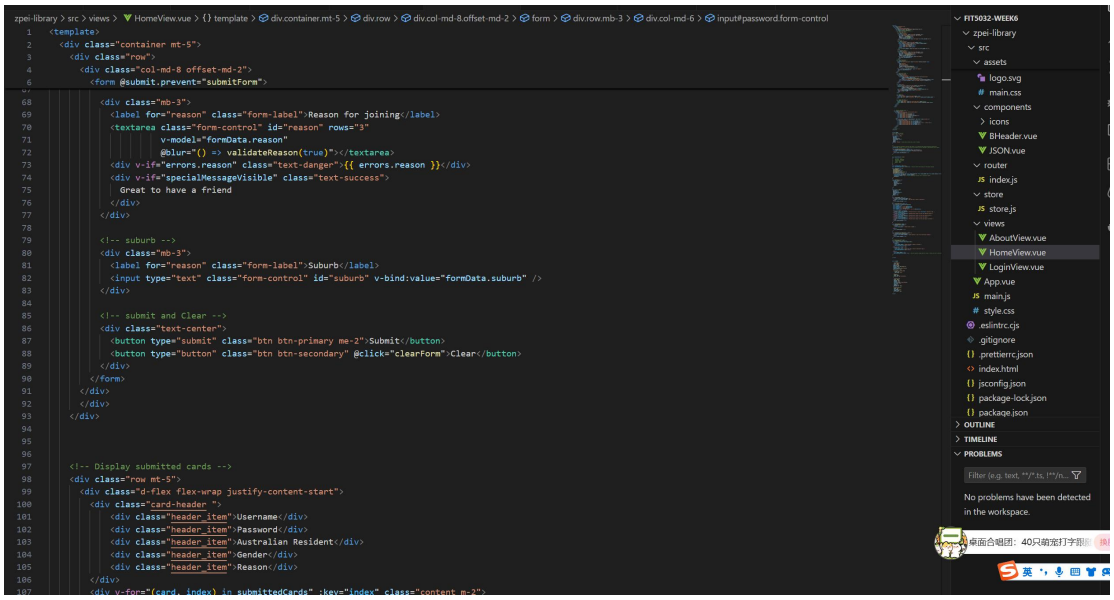
HomeView.vue



```
<template>
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-8 offset-md-2">
        <h1 class="text-center">  W5.Library Registration Form</h1>
        <form @submit.prevent="submitForm">
          <!-- Row for Username and Password -->
          <div class="row mb-3">
            <!-- Username Field -->
            <div class="col-md-6">
              <label for="username" class="form-label">Username</label>
              <input 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>
            <!-- Gender -->
            <div class="col-md-6">
              <label for="gender" class="form-label">Gender</label>
              <select class="form-select" id="gender" v-model="formData.gender">
                <option value="">Select Gender</option>
                <option value="male">Male</option>
                <option value="female">Female</option>
                <option value="other">Other</option>
              </select>
              <div v-if="errors.gender" class="text-danger">{{ errors.gender }}</div>
            </div>
            <!-- Password Field -->
            <div class="col-md-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.confirmPassword" class="text-danger">
                {{ errors.confirmPassword }}
              </div>
            </div>
          </div>
          <!-- confirm password -->
          <div class="col-md-6">
            <label for="confirm-password" class="form-label">Confirm password</label>
            <input
                type="password"
                class="form-control"
                id="confirm-password"
                v-model="formData.confirmPassword"
```



```
<template>
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-8 offset-md-2">
        <form @submit.prevent="submitForm">
            </div>
          <!-- confirm password -->
          <div class="col-md-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>
        <!-- Additional Fields -->
        <div class="row mb-3">
          <div class="col-md-6">
            <div class="form-check">
              <input type="checkbox" class="form-check-input" id="isAustralian"
                     v-model="formData.isAustralian"
                     @blur="() => validateAustralianResident(true)">
              <label class="form-check-label" for="isAustralian">Australian Resident?</label>
              <div v-if="errors.resident" class="text-danger">{{errors.resident}}</div>
            </div>
          </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="() => validateReason(true)"></textarea>
          <div v-if="errors.reason" class="text-danger">{{ errors.reason }}</div>
          <div v-if="specialMessageVisible" class="text-success">
            Great to have a friend
          </div>
        </div>

        <!-- suburb -->
        <div class="mb-3">
```



```
zpei-library > src > views > HomeView.vue > {} template > div.container.mt-5 > div.row > div.col-md-8.offset-md-2 > form > div.row.mb-3 > div.col-md-6 > input#password.form-control
<template>
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-8 offset-md-2">
        <form @submit.prevent="submitForm">
          <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="() => validateReason(true)"></textarea>
            <div v-if="errors.reason" class="text-danger">{{ errors.reason }}</div>
            <div v-if="specialMessageVisible" class="text-success">
              Great to have a friend
            </div>
          </div>

          <!-- suburb -->
          <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>

          <!-- submit and Clear -->
          <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>
        </form>
      </div>
    </div>


    <!-- Display submitted cards -->
    <div class="row mt-5">
      <div class="d-flex flex-wrap justify-content-start">
        <div class="card-header ">
            <div class="header_item">Username</div>
            <div class="header_item">Password</div>
            <div class="header_item">Australian Resident</div>
            <div class="header_item">Gender</div>
            <div class="header_item">Reason</div>
        </div>
        <div v-for="(card, index) in submittedCards" :key="index" class="content m-2">
```

```
<script setup>
const errors = ref({
});

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) =>{
  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 lowcase 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 validateGender =(blur) =>{
  if (!formData.value.gender){
    if(blur){
      errors.value.gender = "Please select a gender.";
    }
  } else {
```