<script setup>

import {computed, onBeforeMount, ref} from 'vue';

import axios from 'axios';

import Cookies from 'js-cookie';

import useUserStore from '../stores/userStore';

import { storeToRefs } from 'pinia';

import { useRouter } from 'vue-router';

onBeforeMount(() => {

  axios.defaults.headers.common['X-CSRFToken'] = Cookies.get("csrftoken");

})

const headline = ref('');

const comment = ref('');

const subject\_id = ref(null);

const fileInput = ref(null);

const subjects = ref([]);

const selectedFiles = ref([]);

const router = useRouter();

const errorText = ref("");

const csrfToken = Cookies.get('csrftoken');

const userStore = useUserStore();

const { userId } = storeToRefs(userStore);

onBeforeMount(async () => {

    await fetchSubjects();

});

async function fetchSubjects() {

    try {

        const response = await axios.get("/api/subjects/");

        subjects.value = response.data;

    } catch (error) {

        console.error("Ошибка при получении предметов:", error);

    }

}

function handleFileChange() {

    const newFiles = Array.from(fileInput.value.files).map(file => {

          let preview = null;

          if(file.type.startsWith('image/')){

              preview= URL.createObjectURL(file)

          }

          else if(file.name.endsWith('.xlsx')){

              preview="/icon-file-xlsx.png"

          }

          else if(file.name.endsWith('.pdf')){

              preview="/icon-file-pdf.png"

          }

          else if (file.name.endsWith('.docx') || file.name.endsWith('.doc')) {

              preview = "/icon-file-word.png";

          }

          else {

              preview="/icon-file-another.png"

          }

        return {file,preview}

        });

      selectedFiles.value.push(...newFiles);

      fileInput.value.value = null

}

function removeFile(index) {

    selectedFiles.value.splice(index, 1);

}

async function addRecord() {

    errorText.value = "";

    const formData = new FormData();

    formData.append('headline', headline.value);

    formData.append('comment', comment.value);

    formData.append('subject\_id', subject\_id.value);

    formData.append('is\_published', false);

    formData.append('author', userId.value);

      for (let i = 0; i < selectedFiles.value.length; i++) {

           formData.append('files', selectedFiles.value[i].file);

        }

    console.log("Data отправляемые на сервер:", formData);

    try {

        const response = await axios.post("/api/records/", formData, {

           headers: {

               'X-CSRFToken': csrfToken,

               'Content-Type': false,

                'processData': false

            }

        });

           if(response.status==201){

               router.push('/records')

          }

       else{

        errorText.value="Произошла ошибка при создании публикации"

       }

    } catch (error) {

        console.error("Ошибка при добавлении записи:", error);

       errorText.value="Произошла ошибка при создании публикации"

    }

}

</script>

<template>

  <div class="add-record-container">

    <h2 class="mb-4 text-white">Добавить новый материал</h2>

         <p v-if="errorText" style="color:red">{{ errorText }}</p>

        <form @submit.prevent="addRecord">

            <div class="mb-3">

                <label for="headline" class="form-label text-white">Заголовок</label>

                <input type="text" class="form-control" id="headline" v-model="headline" required>

            </div>

            <div class="mb-3">

                <label for="comment" class="form-label text-white">Комментарий</label>

                <textarea class="form-control" id="comment" v-model="comment" rows="3"></textarea>

            </div>

            <div class="mb-3">

                <label for="subject" class="form-label text-white">Предметная область</label>

                <select class="form-select" v-model="subject\_id" required>

                    <option :value="null" disabled>Выберите предмет</option>

                    <option v-for="subject in subjects" :key="subject.id" :value="subject.id">{{ subject.name }}</option>

                </select>

            </div>

                <div class="mb-3">

                <label for="fileInput" class="form-label text-white">Файлы</label>

                  <div class="file-input-container">

                      <input type="file" ref="fileInput" class="form-control" id="fileInput" multiple @change="handleFileChange">

                      <ul v-if="selectedFiles.length" class="file-list">

                           <li v-for="(item, index) in selectedFiles" :key="index" class="file-item">

                               <button type="button" class="remove-file-button" @click="removeFile(index)">

                                <i class="bi bi-x-circle"></i>

                                </button>

                              <div class="file-item-content">

                                  <img v-if="item.preview && item.file.type.startsWith('image/')" :src="item.preview" alt="Превью" class="file-preview">

                                  <img v-else-if="item.preview" :src="item.preview" alt="Иконка файла" class="file-icon">

                                 <div class="file-name text-white">{{ item.file.name }}</div>

                              </div>

                           </li>

                      </ul>

                 </div>

                </div>

            <button type="submit" class="btn btn-primary">Добавить</button>

        </form>

    </div>

</template>

<style scoped>

.add-record-container {

    max-width: 600px;

    margin: 20px auto;

    padding: 20px;

    border-radius: 1rem;

    background-color: #343a40;

}

.form-label {

    font-weight: bold;

}

.form-control, .form-select {

  margin-bottom: 1rem;

  background-color: #495057;

   color: white;

   border-color: #495057;

}

.btn-primary {

  background-color: #007bff;

  border-color: #007bff;

    color: white;

}

.btn-primary:hover{

  background-color: #0056b3;

    border-color: #0056b3;

}

.file-input-container{

    position: relative;

}

.file-list{

    list-style: none;

    padding: 0;

    margin-top: 10px;

      display: flex;

    flex-wrap: wrap;

}

.file-item{

  display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: space-between;

    border: 1px solid #eee;

    padding: 10px;

    border-radius: 4px;

    margin-bottom: 5px;

    margin-right: 5px;

     width: 150px;

     position: relative;

}

.file-name{

    font-weight: bold;

     margin-right: 10px;

     color: white;

        word-break: break-word;

         word-wrap: break-word;

}

.remove-file-button{

    background-color: transparent;

    border: none;

    cursor: pointer;

    color: #dc3545;

    font-size: 1.2rem;

    position: absolute;

    top: 5px;

    right: 5px;

}

.remove-file-button:hover{

     color: #a71d2a;

}

.file-preview{

  max-width: 50px;

    max-height: 50px;

    margin-right: 10px;

}

.file-item-content{

  display: flex;

    flex-direction: column;

    align-items: center;

    text-align: center;

    margin-bottom: 5px;

}

.file-icon{

  max-width: 50px;

    max-height: 50px;

    margin-right: 10px;

    margin-bottom: 5px;

}

</style>