

Code source - PWA

Jonathan Borel-Jaquet

10 juin 2021

Listings

1	./Sources/src/App.vue	3
2	./Sources/src/components/About.vue	3
3	./Sources/src/components/Administration.vue	4
4	./Sources/src/components/ButtonReturn.vue	11
5	./Sources/src/components/Calendar.vue	12
6	./Sources/src/components/Connection.vue	18
7	./Sources/src/components/CustomerAppointment.vue	20
8	./Sources/src/components/CustomerInformation.vue	29
9	./Sources/src/components/EducatorCalendar.vue	51
10	./Sources/src/components/Footer.vue	56
11	./Sources/src/components/Home.vue	57
12	./Sources/src/components/Inscription.vue	58
13	./Sources/src/components/LeftSectionContent.vue	62
14	./Sources/src/components/Navbar.vue	64
15	./Sources/src/components/Planning.vue	65
16	./Sources/src/components/PrivacyPolicy.vue	84
17	./Sources/src/components/RightSectionContent.vue	86
18	./Sources/src/components/SignaturePad.vue	88
19	./Sources/src/constants.js	90
20	./Sources/src/main.js	90
21	./Sources/src/plugins/bootstrap-vue.js	91
22	./Sources/src/plugins/vue-alertify.js	91
23	./Sources/src/plugins/vue-signature-pad.js	91
24	./Sources/src/router.js	91
25	./Sources/src/store.js	94

Listing 1 – ./Sources/src/App.vue

```

1  <!--
2    App.vue
3
4    Main root component of the application also serving as a template for the
      application.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div id="app">
11      <div id="content">
12        <Navbar />
13        <router-view />
14      </div>
15      <Footer />
16    </div>
17  </template>
18
19  <script>
20  import Footer from "../components/Footer.vue";
21  import Navbar from "../components/Navbar.vue";
22
23  export default {
24    name: "app",
25    components: {
26      Navbar,
27      Footer,
28    },
29    created() {
30      this.$store.dispatch("autoLogin");
31    },
32  };
33  </script>
34
35  <style>
36  #content {
37    min-height: calc(100vh - 260px);
38  }
39  </style>

```

Listing 2 – ./Sources/src/components/About.vue

```

1  <!--
2    About.vue
3
4    Component representing the page about of the application.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div>
11      <left-section-content
12        :textH1="textH1LeftSectionContent"
13        :textH5="textH5LeftSectionContent"
14        :textParagraph="textParagraphLeftSectionContent"
15        :imageSource="imageSourceLeftSectionContent"

```

```

16     ></left-section-content>
17     <right-section-content
18       :textH5="textH5RightSectionContent"
19       :textParagraph="textParagraphRightSectionContent"
20       :routeBouton="routeBoutonRightSectionContent"
21       :imageSource="imageSourceRightSectionContent"
22     ></right-section-content>
23   </div>
24 </template>
25
26 <script>
27 import LeftSectionContent from "../LeftSectionContent.vue";
28 import BorisEducator from "../assets/img/boriseducateurcanin.png";
29 import RightSectionContent from "../RightSectionContent.vue";
30 import BorisWithHyron from "../assets/img/boriswithhyron.png";
31
32 export default {
33   components: { LeftSectionContent, RightSectionContent },
34   name: "About",
35   data() {
36     return {
37       texth1LeftSectionContent: "Bonjour, je suis Boris Gourdoux",
38       texth5LeftSectionContent: "+ de 300 couples Maître/chien à mon actif",
39       textParagraphLeftSectionContent:
40         "Depuis ma toute petite enfance, ma relation avec le chien à été salvatrice
          . Elle m'a permis de dépasser bien des difficultés et de trouver une vé
          ritable complicité. Diplômé et spécialisé en rééducation et
          comportement, vous bénéficiez de toute mon expertise et expérience dans
          l'éducation canine, fruit de nombreuses formations dans ce qui se fait
          de mieux dans le domaine. L'objectif chez Douceur de Chien est de vous
          permettre de créer un lien renforcé et durable avec votre compagnon à
          4 pattes et ceci en utilisant uniquement de la patience, de la cohé
          rence et beaucoup de passion. Ainsi chez Douceur de Chien vous n'
          obtiendrez aucun conseil sur l'utilisation de colliers étrangleurs ou m
          ême de la force pour éduquer votre chien.",
41       imageSourceLeftSectionContent: BorisEducator,
42
43       texth5RightSectionContent: "Une méthode moderne",
44       textParagraphRightSectionContent:
45         "Je base ma méthode en respectant les besoins et l'émotionnel du chien, c'
          est pourquoi je met l'accent sur une coopération totale de ce dernier
          en lui permettant de faire ses propres choix et ainsi vous permettre d'
          obtenir de votre chien, écoute et attention en toutes circonstances.",
46       imageSourceRightSectionContent: BorisWithHyron,
47       routeBoutonRightSectionContent: "about",
48     };
49   },
50 };
51 </script>
52
53 <style scoped></style>

```

Listing 3 – ./Sources/src/components/Administration.vue

```

1 <!--
2   Administration.vue
3
4   Component representing the administration page of the canine educator of the
    application.

```

```

5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7 -->
8
9 <template>
10 <div>
11   <b-container>
12     <b-row class="title text-center">
13       <b-col>
14         <h1>Administration</h1>
15       </b-col>
16     </b-row>
17     <b-row>
18       <b-col>
19         <b-button
20           id="btnUpdatePassword"
21           variant="outline-primary"
22           v-b-modal.modal-update-password
23           block
24         >
25           Modifier mon mot de passe
26         </b-button>
27         <b-button
28           id="btnAddClient"
29           style="margin-bottom: 10px; margin-top: 10px"
30           variant="outline-primary"
31           block
32           v-b-modal.modal-add-client
33         >
34           Ajouter un client
35         </b-button>
36       </b-col>
37     </b-row>
38     <b-row>
39       <b-col>
40         <b-input-group style="margin-bottom: 10px">
41           <b-form-input
42             id="filter-input"
43             v-model="filter"
44             type="search"
45             placeholder="Tapez pour rechercher"
46           ></b-form-input>
47
48           <b-input-group-append>
49             <b-button :disabled="!filter" @click="filter = ''">
50               Effacer
51             </b-button>
52           </b-input-group-append>
53         </b-input-group>
54       </b-col>
55     </b-row>
56     <b-pagination
57       v-model="btableOptions.currentPage"
58       align="fill"
59       size="sm"
60       class="my-0"
61       :total-rows="totalRowsPagination"
62     ></b-pagination>
63   </b-table>

```

```

64     ref="tableuser"
65     :items="btableOptions.items"
66     :busy="isBusy"
67     :fields="btableOptions.fields"
68     :sort-by.sync="btableOptions.sortBy"
69     :sort-desc.sync="btableOptions.sortDesc"
70     :filter="filter"
71     :filter-included-fields="filterOn"
72     :perPage="10"
73     :current-page="btableOptions.currentPage"
74     sort-icon-left
75     striped
76     responsive="sm"
77 >
78     <template #table-busy>
79         <div class="text-center text-danger my-2">
80             <b-spinner class="align-middle"></b-spinner>
81             <strong> Chargement...</strong>
82         </div>
83     </template>
84
85     <template #cell(chiens)="row">
86         <b-button size="sm" @click="row.toggleDetails" class="mr-2 tableBtn">
87             {{ row.detailsShowing ? "Cacher" : "Afficher" }}
88         </b-button>
89     </template>
90     <template #cell(edition)="row">
91         <b-button
92             :to="{
93                 name: 'customerInformation',
94                 params: { userId: row.item.id },
95             }"
96             class="tableBtn"
97         >
98             <b-icon-pencil-square></b-icon-pencil-square>
99         </b-button>
100     </template>
101     <template #row-details="row">
102         <div v-if="row.item.dogs.length > 0">
103             <b-card
104                 style="margin-bottom: 5px"
105                 v-for="dog in row.item.dogs"
106                 :key="dog.id"
107             >
108                 <b-row>
109                     <b-col><b>Nom du chien:</b></b-col>
110                     <b-col>{{ dog.name }}</b-col>
111                 </b-row>
112                 <b-row>
113                     <b-col><b>Photo du chien:</b></b-col>
114                     <b-col v-if="dog.picture_serial_id">
115                         <b-img
116                             :src="
117                                 $API_URL + 'dogs/downloadPicture/' + dog.picture_serial_id
118                             "
119                             rounded
120                             fluid
121                             :alt="dog.name + ' picture'"
122                         ></b-img>

```

```

123         </b-col>
124         <b-col v-else> Le chien n'a pas encore de photo</b-col>
125     </b-row>
126 </b-card>
127 </div>
128 <b-card v-else>
129     <b-row>
130         <b-col>L'utilisateur n'a aucun chiens pour l'instant</b-col>
131     </b-row>
132 </b-card>
133 </template>
134 </b-table>
135
136 <!-- MODAL ADD CLIENT -->
137 <b-modal
138     id="modal-add-client"
139     title="Ajouter un client"
140     :hide-footer="true"
141 >
142     <b-form
143         @submit.prevent="
144             createUser(
145                 formAddClient.email,
146                 formAddClient.firstname,
147                 formAddClient.lastname,
148                 formAddClient.phonenumber,
149                 formAddClient.address
150             )
151         "
152     >
153         <b-form-group
154             id="input-group-client-email"
155             label="Adresse e-mail :"
156             label-for="input-client-email"
157         >
158             <b-form-input
159                 id="input-client-email"
160                 v-model="formAddClient.email"
161                 type="email"
162                 placeholder="Entrez l'adresse e-mail du client"
163                 required
164             ></b-form-input>
165         </b-form-group>
166         <b-form-group
167             id="input-group-client-firstname"
168             label="Prénom :"
169             label-for="input-client-firstname"
170         >
171             <b-form-input
172                 id="input-client-firstname"
173                 v-model="formAddClient.firstname"
174                 type="text"
175                 placeholder="Entrez le prénom du client"
176                 required
177             ></b-form-input>
178         </b-form-group>
179         <b-form-group
180             id="input-group-client-lastname"
181             label="Nom de famille :"

```

```

182     label-for="input-client-lastname"
183   >
184     <b-form-input
185       id="input-client-lastname"
186       v-model="formAddClient.lastname"
187       type="text"
188       placeholder="Entrez le nom de famille du client"
189       required
190     ></b-form-input>
191   </b-form-group>
192   <b-form-group
193     id="input-group-client-phonenummer"
194     label="Numéro de téléphone :"
195     label-for="input-client-phonenummer"
196   >
197     <b-form-input
198       id="input-client-phonenummer"
199       v-model="formAddClient.phonenummer"
200       type="text"
201       placeholder="Entrez le numéro de téléphone du client"
202       required
203     ></b-form-input>
204   </b-form-group>
205   <b-form-group
206     id="input-group-client-address"
207     label="Adresse :"
208     label-for="input-client-address"
209   >
210     <b-form-input
211       id="input-client-address"
212       v-model="formAddClient.address"
213       type="text"
214       placeholder="Entrez l'adresse du client"
215       required
216     ></b-form-input>
217   </b-form-group>
218   <b-button block type="submit" variant="outline-primary">
219     Ajouter le client
220   </b-button>
221 </b-form>
222 </b-modal>
223
224 <!-- MODAL UPDATE PASSWORD -->
225 <b-modal
226   id="modal-update-password"
227   title="Modifier le mot de passe"
228   :hide-footer="true"
229 >
230   <b-form
231     @submit.prevent="updateAuthCustomerPassword(password, repeatPassword)"
232   >
233     <b-form-group
234       id="input-group-inscription-password"
235       label="Mot de passe :"
236       label-for="input-inscription-password"
237     >
238       <b-form-input
239         id="input-inscription-password"
240         v-model="formChangePassword.password"

```



```

241         type="password"
242         placeholder="Entrez le mot de passe"
243         required
244     ></b-form-input>
245 </b-form-group>
246 <b-form-group
247     id="input-group-inscription-second-password"
248     label="Confirmation du mot de passe :"
249     label-for="input-inscription-second-password"
250 >
251     <b-form-input
252         id="input-inscription-second-password"
253         v-model="formChangePassword.repeatPassword"
254         type="password"
255         placeholder="Entrez le mot de passe"
256         required
257     ></b-form-input>
258 </b-form-group>
259
260     <b-button block type="submit" variant="outline-primary">
261         Modifier le mot de passe
262     </b-button>
263 </b-form>
264 </b-modal>
265 </b-container>
266 </div>
267 </template>
268
269 <script>
270 import { BIconPencilSquare } from "bootstrap-vue";
271
272 export default {
273     components: {
274         BIconPencilSquare,
275     },
276     name: "Administration",
277     data() {
278         return {
279             btableOptions: {
280                 sortBy: "nom",
281                 sortDesc: false,
282                 fields: [
283                     { key: "nom", sortable: true },
284                     { key: "prenom", sortable: true },
285                     { key: "chiens", sortable: false },
286                     { key: "edition", sortable: false },
287                 ],
288                 items: [],
289                 currentPage: 1,
290             },
291             formAddClient: {
292                 email: "",
293                 firstname: "",
294                 lastname: "",
295                 phonenumber: "",
296                 address: "",
297             },
298             formChangePassword: {
299                 password: "",

```

```

300         repeatPassword: "",
301     },
302     filter: null,
303     filterOn: ["nom", "prenom"],
304     isBusy: true,
305     totalRowsPagination: 1,
306 };
307 },
308 methods: {
309     /**
310      * Method to load all customers with their dogs from the api rest endpoint "GET
311      * api/v1/users".
312      */
313     loadCustomersWithDogs() {
314         const config = {
315             headers: {
316                 "Authorization" : this.$store.state.api_token
317             },
318         };
319         this.$http
320             .get(this.$API_URL + "users/", config)
321             .then((response) => {
322                 this.btableOptions.items = [];
323                 response.data.forEach((user) => {
324                     user["prenom"] = user["firstname"];
325                     delete user["firstname"];
326                     user["nom"] = user["lastname"];
327                     delete user["lastname"];
328                 });
329                 this.btableOptions.items = response.data;
330                 this.toggleBusy();
331                 this.totalRowsPagination = this.btableOptions.items.length + 10;
332             })
333             .catch((error) => {
334                 this.$alertify.error(error.response.data.error);
335             });
336     },
337     /**
338      * Method to create a user with the api rest endpoint "POST api/v1/users".
339      *
340      * @param {string} email The user's email address
341      * @param {string} firstname The user's first name
342      * @param {string} lastname The user's last name
343      * @param {string} phonenumber The user's phonenumber
344      * @param {string} address The user's address
345      */
346     createUser(email, firstname, lastname, phonenumber, address) {
347         const params = new URLSearchParams();
348         params.append("email", email);
349         params.append("firstname", firstname);
350         params.append("lastname", lastname);
351         params.append("phonenumber", phonenumber);
352         params.append("address", address);
353         const config = {
354             headers: {
355                 "Authorization" : this.$store.state.api_token,
356             },
357         };

```

```

358     this.$http
359       .post(this.$API_URL + "users/", params, config)
360       .then((response) => {
361         console.log(response);
362         this.loadCustomersWithDogs();
363         this.$alertify.success("Client ajouté avec succès");
364         this.$bvModal.hide("modal-add-client");
365       })
366       .catch((error) => {
367         this.$alertify.error(error.response.data.error);
368       });
369   },
370   /**
371    * Method to update the password of the authenticated user with the api rest
372    * endpoint "PATCH api/v1/users/me/changePassword".
373    *
374    * @param {string} password The new password
375    * @param {string} repeatPassword Repetition of the new password
376    */
377   updateAuthCustomerPassword(password, repeatPassword) {
378     if (password !== repeatPassword) {
379       this.$alertify.error("les mots de passes ne correspondent pas");
380       return;
381     }
382     const params = new URLSearchParams();
383     params.append("password", password);
384     const config = {
385       headers: {
386         "Authorization" : this.$store.state.api_token,
387       },
388     };
389     this.$http
390       .patch(this.$API_URL + "users/me/changePassword/", params, config)
391       .then((response) => {
392         console.log(response);
393         this.$alertify.success("Mot de passe modifié");
394         this.$bvModal.hide("modal-update-password");
395       })
396       .catch((error) => {
397         this.$alertify.error(error.response.data.error);
398       });
399   },
400   toggleBusy() {
401     this.isBusy = false;
402   },
403   mounted() {
404     this.loadCustomersWithDogs();
405   },
406 };
407 </script>
408
409 <style scoped>
410 .tableBtn {
411   background-color: #3ea3d8;
412 }
413 </style>

```

Listing 4 – ./Sources/src/components/ButtonReturn.vue

```

1  <!--
2    ButtonReturn.vue
3
4    Reusable component representing a static button on a page allowing to redirect
      the user to another component.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div>
11      <b-button class="btnReturn" :to="to">
12        <p class="h4">
13          <b-icon-arrow-return-left></b-icon-arrow-return-left>
14        </p>
15      </b-button>
16    </div>
17  </template>
18
19  <script>
20  import { BIconArrowReturnLeft } from "bootstrap-vue";
21
22  export default {
23    components: {
24      BIconArrowReturnLeft,
25    },
26    name: "ButtonReturn",
27    props: {
28      to: Object,
29    },
30  };
31  </script>
32
33  <style scoped>
34  .btnReturn {
35    padding-top: 18px;
36    position: fixed;
37    bottom: 100px;
38    right: 15px;
39    z-index: 5;
40    display: block;
41    height: 70px;
42    width: 70px;
43    border-radius: 50%;
44    background-color: #008afc;
45    box-shadow: -1px -1px 15px 1px rgba(0, 0, 0, 0.7);
46  }
47  </style>

```

Listing 5 – ./Sources/src/components/Calendar.vue

```

1  <!--
2    Calendar.vue
3
4    Component representing the page of the calendars of the dog educators of the
      society.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>

```

```

7  -->
8
9  <template>
10  <div>
11    <b-container>
12      <b-row id="title" class="text-center">
13        <b-col>
14          <h1>Agenda des éducateurs canins de la société</h1>
15        </b-col>
16      </b-row>
17      <b-row id="title" class="text-center">
18        <b-col>
19          <b-form-group label="Éducateur canin :">
20            <b-form-select
21              id="educatorSelect"
22              v-model="idEducatorSelected"
23              @change="onChange()"
24            >
25              <b-form-select-option
26                v-for="educator in educators"
27                :key="educator.id"
28                :value="educator.id"
29                >{{ educator.firstname }}
30                {{ educator.lastname }}</b-form-select-option
31              >
32            </b-form-select>
33          </b-form-group>
34        </b-col>
35      </b-row>
36      <b-row class="text-center">
37        <b-col>
38          <p class="font-weight-light">
39            Vous voulez prendre un rendez-vous avec un des éducateurs canins de
40            la société ? <br />
41            Il vous suffit de cliquer sur un des créneaux horaires une fois
42            connecté afin de réserver votre rendez-vous
43          </p>
44        </b-col>
45      </b-row>
46      <b-row class="text-center">
47        <b-col>
48          <b-badge id="badge-free" variant="success">Rendez-vous libre</b-badge>
49        </b-col>
50      </b-row>
51      <b-row class="text-center">
52        <b-col>
53          <FullCalendar ref="fullCalendar" :options="calendarOptions" />
54        </b-col>
55      </b-row>
56      <b-modal
57        id="modal-create-appointment"
58        title="Réserver un rendez-vous"
59        :hide-footer="true"
60      >
61        <div v-if="auth">
62          <b-form
63            @submit.prevent="
64              createAppointment(
65                selectedAppointment.datetime,

```

```

66         selectedAppointment.durationInHour,
67         $store.state.user_id,
68         selectedAppointment.idEducator
69     )
70     "
71 >
72     <b-row>
73         <b-col class="d-flex justify-content-center">
74             <h5>Éducateur canin</h5>
75         </b-col>
76     </b-row>
77     <b-row>
78         <b-col class="d-flex justify-content-center">
79             <p>
80                 {{ this.$jquery("#educatorSelect option:selected").text() }}
81             </p>
82         </b-col>
83     </b-row>
84     <b-row>
85         <b-col class="d-flex justify-content-center">
86             <h5>Date</h5>
87         </b-col>
88     </b-row>
89     <b-row>
90         <b-col>
91             <b-calendar
92                 block
93                 id="date-calendar"
94                 label-help=""
95                 :readonly="true"
96                 :value="selectedAppointment.datetime"
97             ></b-calendar>
98         </b-col>
99     </b-row>
100    <b-row>
101        <b-col class="d-flex justify-content-center">
102            <h5>Créneau horaire</h5>
103        </b-col>
104    </b-row>
105    <b-row>
106        <b-col class="d-flex justify-content-center">
107            <p>
108                {{ selectedAppointment.startHour }}h à
109                {{ selectedAppointment.endHour }}h
110            </p>
111        </b-col>
112    </b-row>
113    <b-button block type="submit" variant="outline-primary">
114        Réserver le rendez-vous
115    </b-button>
116 </b-form>
117 </div>
118 <div v-else>
119     <p>Connectez-vous afin de pouvoir planifier votre rendez-vous.</p>
120     <b-button to="/connection" variant="primary">Se connecter</b-button>
121 </div>
122 </b-modal>
123 </b-container>
124 </div>

```

```

125 </template>
126
127 <script>
128 import FullCalendar from "@fullcalendar/vue";
129 import dayGridPlugin from "@fullcalendar/daygrid";
130 import timeGridPlugin from "@fullcalendar/timegrid";
131 import interactionPlugin from "@fullcalendar/interaction";
132 import frLocale from "@fullcalendar/core/locales/fr";
133 import moment from "moment";
134 moment.locale("fr-ch");
135
136 export default {
137   components: {
138     FullCalendar,
139   },
140   name: "Calendar",
141   data() {
142     return {
143       calendarOptions: {
144         plugins: [dayGridPlugin, timeGridPlugin, interactionPlugin],
145         initialView: "dayGridMonth",
146         headerToolbar: {
147           left: "prev,next today",
148           center: "title",
149           right: "dayGridMonth,timeGridWeek,timeGridDay",
150         },
151         height: "auto",
152         locale: frLocale,
153         eventDisplay: "block",
154         allDaySlot: false,
155         slotMinTime: "06:00:00",
156         slotMaxTime: "20:00:00",
157         events: [],
158         eventBackgroundColor: "green",
159         eventClick: this.handleDateClick,
160       },
161       selectedAppointment: {
162         datetime: "",
163         startHour: "",
164         endHour: "",
165         durationInHour: null,
166         idEducator: null,
167       },
168       idEducatorSelected: null,
169       educators: [],
170     };
171   },
172   methods: {
173     /**
174      * Method to load all educators from the api rest endpoint "GET api/v1/users/
175      * educators".
176      */
177     loadEducators() {
178       this.$http.get(this.$API_URL + "users/educators/").then((response) => {
179         this.educators = response.data;
180         this.idEducatorSelected = this.educators[0].id;
181         this.loadEducatorPlanning(this.idEducatorSelected);
182       });
183     }
184   }
185 }

```

```

183   },
184   /**
185    * Method to load all the time slots of an educator from his id from the api
      rest endpoint "GET api/v1/plannings/{idEducator}".
186    *
187    * @param {number} idEducator The dog educator's id
188    */
189   loadEducatorPlanning(idEducator) {
190     this.$http
191       .get(this.$API_URL + "plannings/" + idEducator)
192       .then((response) => {
193         this.calendarOptions.events = [];
194         response.data.forEach((event) => {
195           this.calendarOptions.events.push({
196             title: "Disponible",
197             start: event.date + " " + event.time_start,
198             end: event.date + " " + event.time_end,
199             extendedProps: {
200               idEducator: idEducator,
201             },
202           });
203         });
204       })
205       .catch((error) => {
206         this.$alertify.error(error.response.data.error);
207       });
208   },
209   /**
210    * Method called when clicking on a time slot in the calendar to redirect the
      user to the daily overview and open the modal for creating appointments.
211    *
212    * @param {number} info The object of FullCalendar
213    */
214   handleDateClick: function (info) {
215     {
216       let calendarApi = this.$refs.fullCalendar.getApi();
217       if (info.view.type == "dayGridMonth") {
218         calendarApi.gotoDate(info.event.endStr);
219         calendarApi.changeView("timeGridDay");
220       }
221       if (
222         info.view.type == "timeGridWeek" ||
223         info.view.type == "timeGridDay"
224       ) {
225         var startDate = moment(info.event.start);
226         var endDate = moment(info.event.end);
227         var datetime = startDate.format("YYYY-MM-DD HH:mm:ss");
228         var startHour = startDate.format("HH");
229         var endHour = endDate.format("HH");
230         var durationDifference = moment.duration(
231           moment(info.event.end).diff(moment(info.event.start))
232         );
233         var durationInHour = durationDifference.asHours();
234         this.sendAppointmentInformations(
235           datetime,
236           startHour,
237           endHour,
238           durationInHour,
239           info.event.extendedProps.idEducator

```



```

240         );
241         this.$bvModal.show("modal-create-appointment");
242     }
243 }
244 },
245 /**
246  * Method to create an appointment with the api rest endpoint "POST api/v1/
    appointments".
247  *
248  * @param {string} datetime The datetime of the appointment
249  * @param {number} durationInHour The duration in hours of the appointment
250  * @param {number} user_id_customer The customer's identifier
251  * @param {number} user_id_educator The dog educator's identifier
252  */
253 createAppointment(
254     datetime,
255     durationInHour,
256     user_id_customer,
257     user_id_educator
258 ) {
259     const params = new URLSearchParams();
260     params.append("datetime_appointment", datetime);
261     params.append("duration_in_hour", durationInHour);
262     params.append("user_id_customer", user_id_customer);
263     params.append("user_id_educator", user_id_educator);
264     const config = {
265         headers: {
266             "Authorization" : this.$store.state.api_token,
267         },
268     };
269     this.$http
270     .post(this.$API_URL + "appointments/", params, config)
271     .then((response) => {
272         console.log(response);
273         this.$alertify.success("Rendez-vous réservé avec succès");
274         this.$bvModal.hide("modal-create-appointment");
275         this.loadEducatorPlanning(this.idEducatorSelected);
276     })
277     .catch((error) => {
278         this.$alertify.error(error.response.data.error);
279     });
280 },
281 /**
282  * Method to load another calendar when changing the selection of a dog
    educator
283  *
284  */
285 onChange() {
286     this.loadEducatorPlanning(this.idEducatorSelected);
287 },
288 sendAppointmentInformations(
289     datetime,
290     startHour,
291     endHour,
292     durationInHour,
293     idEducator
294 ) {
295     this.selectedAppointment.datetime = datetime;
296     this.selectedAppointment.startHour = startHour;

```

```

297     this.selectedAppointment.endHour = endHour;
298     this.selectedAppointment.durationInHour = durationInHour;
299     this.selectedAppointment.idEducator = idEducator;
300   },
301 },
302 mounted() {
303   this.loadEducators();
304 },
305 computed: {
306   auth() {
307     return this.$store.getters.isAuthenticated;
308   },
309 },
310 };
311 </script>
312
313 <style scoped>
314 #badge-free {
315   padding: 10px;
316   margin-bottom: 10px;
317   background-color: #008000;
318   border: #378d8d 1px solid;
319 }
320 #title {
321   margin-top: 20px;
322   color: #3ea3d8;
323 }
324 </style>

```

Listing 6 – ./Sources/src/components/Connection.vue

```

1  <!--
2    Connection.vue
3
4    Component representing the application connection page.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div class="content">
11      <b-container>
12        <b-row id="title" class="text-center">
13          <b-col>
14            <h1>Se connecter</h1>
15          </b-col>
16        </b-row>
17        <b-row class="text-center text-secondary">
18          <b-col>
19            Nouveau sur ce site ?
20            <router-link to="/inscription">
21              <span role="link">S'inscrire</span>
22            </router-link>
23          </b-col>
24        </b-row>
25        <b-row id="title" class="justify-content-center">
26          <b-col lg="6">
27            <b-form @submit.prevent="connection" autocomplete="on">
28              <b-form-group>

```

```

29         id="input-group-connection-email"
30         label="Adresse e-mail:"
31         label-for="input-connection-email"
32     >
33         <b-form-input
34             id="input-connection-email"
35             v-model="form.email"
36             type="email"
37             placeholder="Entrez l'adresse e-mail"
38             autocomplete="email"
39             required
40         ></b-form-input>
41     </b-form-group>
42     <b-form-group
43         id="input-group-connection-password"
44         label="Mot de passe :"
45         label-for="input-connection-password"
46     >
47         <b-form-input
48             id="input-connection-password"
49             v-model="form.password"
50             type="password"
51             placeholder="Entrez le mot de passe"
52             autocomplete="current-password"
53             required
54         ></b-form-input>
55     </b-form-group>
56     <b-button block type="submit" variant="outline-primary">
57         Connexion
58     </b-button>
59 </b-form>
60 </b-col>
61 </b-row>
62 </b-container>
63 </div>
64 </template>
65
66 <script>
67 export default {
68     name: "Connection",
69     data() {
70         return {
71             form: {
72                 email: "",
73                 password: "",
74             },
75         };
76     },
77     methods: {
78         /**
79          * Method to connect the user with the store.
80          *
81          */
82         connection() {
83             this.$store.dispatch("login", {
84                 email: this.form.email,
85                 password: this.form.password,
86             });
87         },

```

```

88     },
89   };
90   </script>
91
92   <style scoped>
93   #title {
94     margin-top: 20px;
95     color: #3ea3d8;
96   }
97   </style>

```

Listing 7 – ./Sources/src/components/CustomerAppointment.vue

```

1  <!--
2    CustomerAppointment.vue
3
4    Component representing the appointment information of a client.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10   <div class="ComponnentContent">
11     <b-container>
12       <b-row class="text-center">
13         <b-col>
14           <h1>{{ title }} {{ customer.firstname }} {{ customer.lastname }}</h1>
15         </b-col>
16       </b-row>
17       <b-row v-if="appointments.length > 0" class="content">
18         <b-col>
19           <div class="card">
20             <div class="card-body">
21               <div id="content">
22                 <ul class="timeline">
23                   <li
24                     class="event"
25                     v-for="appointment in appointments"
26                     :key="appointment.id"
27                     :data-date="
28                       appointment.hourStart + ' - ' + appointment.hourEnd
29                   ">
30                     <h4>{{ appointment.date }}</h4>
31                     <h3>Éducateur canin</h3>
32                     <p>{{ appointment.educatorFullname }}</p>
33                     <h3>Résumé du rendez-vous</h3>
34                     <p v-if="appointment.summary != null">
35                       {{ appointment.summary }}
36                     </p>
37                     <p v-else>Aucun résumé</p>
38                     <div v-if="authAdministrator">
39                       <div style="position: absolute; top: 5px; right: 5px">
40                         <b-button
41                           variant="danger"
42                           v-b-modal.modal-delete-appointment
43                           @click="
44                             sendAppointmentInformations(
45                               appointment.id,

```

```

47         appointment.summary,
48         appointment.noteText,
49         appointment.noteGraphicalSerialId
50     )
51     "
52     >
53     <b-icon-trash id="lol"></b-icon-trash>
54     </b-button>
55 </div>
56 <h3>Notes textuelles personnelles du rendez-vous</h3>
57 <p v-if="appointment.noteText != null">
58     {{ appointment.noteText }}
59 </p>
60 <p v-else>Aucunes notes textuelles</p>
61 <b-button
62     id="btnGraphicalNote"
63     variant="outline-primary"
64     block
65     v-b-modal.modal-graphical-note
66     @click="
67         sendAppointmentInformations(
68             appointment.id,
69             appointment.summary,
70             appointment.noteText,
71             appointment.noteGraphicalSerialId
72         )
73     "
74     >
75     Notes graphiques
76 </b-button>
77 <b-button
78     id="btnTextualNote"
79     variant="outline-primary"
80     block
81     v-b-modal.modal-update-appointment-informations
82     @click="
83         sendAppointmentInformations(
84             appointment.id,
85             appointment.summary,
86             appointment.noteText,
87             appointment.noteGraphicalSerialId
88         )
89     "
90     >
91     Notes textuelles
92 </b-button>
93 </div>
94 </li>
95 </ul>
96 </div>
97 </div>
98 </div>
99 </b-col>
100 </b-row>
101 <b-row v-else class="content">
102     <b-col>
103         {{ messageNoAppointments }}
104     </b-col>
105 </b-row>

```

```

106 <button-return
107   v-if="authAdministrator"
108   :to="{
109     name: 'customerInformation',
110     params: { userId: $route.params.userId },
111   }"
112 ></button-return>
113
114 <!-- MODAL UPDATE APPOINTMENT INFORMATIONS-->
115 <b-modal
116   id="modal-update-appointment-informations"
117   title="Modifier le notes du rendez-vous"
118   :hide-footer="true"
119 >
120   <b-form
121     @submit.prevent="
122       updateAppointmentById(
123         selectedAppointment.id,
124         selectedAppointment.summary,
125         selectedAppointment.noteText
126       )
127     "
128   >
129     <b-form-group
130       id="input-group-appointment-summary"
131       label="Résumé :"
132       label-for="input-appointment-summary"
133     >
134       <b-form-textarea
135         id="input-appointment-summary"
136         v-model="selectedAppointment.summary"
137         placeholder="Entrer le résumé du cours"
138         rows="8"
139         max-rows="8"
140       ></b-form-textarea>
141     </b-form-group>
142
143     <b-form-group
144       id="input-group-appointment-note-text"
145       label="Notes textuelles personnelles :"
146       label-for="input-appointment-note-text"
147     >
148       <b-form-textarea
149         id="input-appointment-note-text"
150         v-model="selectedAppointment.noteText"
151         placeholder="Entrer les notes textuelles du cours"
152         rows="8"
153         max-rows="8"
154       ></b-form-textarea>
155     </b-form-group>
156
157     <b-button block type="submit" variant="outline-primary">
158       Modifier les notes
159     </b-button>
160   </b-form>
161 </b-modal>
162
163 <!-- MODAL GRAPHICAL NOTE-->
164 <b-modal

```

```

165     id="modal-graphical-note"
166     modal-class="modal-fullscreen"
167     title="Notes graphiques"
168     :hide-footer="true"
169     @shown="
170         $refs.signaturePad.resizeCanvas();
171         loadNoteGraphicalPicture(selectedAppointment.noteGraphicalSerialId);
172     "
173 >
174     <b-form
175         style="width: 100%"
176         @submit.prevent="
177             uploadNoteGraphicalByAppointmentId(
178                 selectedAppointment.id,
179                 $refs.signaturePad.saveSignature('image/jpeg').data
180             )
181         "
182     >
183         <VueSignaturePad
184             id="signature"
185             width="100%"
186             height="80%"
187             :customStyle="customStyleSignaturePad"
188             ref="signaturePad"
189             :options="{ backgroundColor: 'rgb(255, 255, 255)' }"
190         />
191         <b-row style="margin-top: 10px">
192             <b-col>
193                 <b-button
194                     block
195                     @click="clearNoteGraphical"
196                     variant="outline-danger"
197                 >
198                     Effacer
199                 </b-button>
200             </b-col>
201             <b-col>
202                 <b-button block type="submit" variant="outline-success">
203                     Sauvegarder
204                 </b-button>
205             </b-col>
206         </b-row>
207     </b-form>
208 </b-modal>
209
210 <!-- MODAL DELETE APPOINTMENT -->
211 <b-modal
212     id="modal-delete-appointment"
213     title="Confirmation de suppression d'un rendez-vous"
214     :hide-footer="true"
215     :hide-header="true"
216 >
217     <h5 style="text-align: center">Supprimer le rendez-vous ?</h5>
218     <b-form @submit.prevent="deleteAppointmentById(selectedAppointment.id)">
219         <b-row>
220             <b-col>
221                 <b-button
222                     block
223                     variant="danger"

```

```

224         @click="$bvModal.hide('modal-delete-appointment')"
225     >
226         Non
227     </b-button>
228 </b-col>
229 <b-col>
230     <b-button
231         block
232         type="submit"
233         variant="success"
234         @click="$bvModal.hide('modal-delete-appointment')"
235     >
236         Oui
237     </b-button>
238 </b-col>
239 </b-row>
240 </b-form>
241 </b-modal>
242 </b-container>
243 </div>
244 </template>
245
246 <script>
247 import ButtonReturn from "./ButtonReturn.vue";
248 import "@/assets/css/timeline.css";
249 import { BIconTrash } from "bootstrap-vue";
250 export default {
251     name: "CustomerAppointment",
252     components: {
253         ButtonReturn,
254         BIconTrash,
255     },
256     data() {
257         return {
258             title: "",
259             customer: {
260                 firstname: "",
261                 lastname: "",
262             },
263             selectedAppointment: {
264                 id: null,
265                 summary: "",
266                 noteText: "",
267                 noteGraphicalSerialId: "",
268             },
269             messageNoAppointments: "Aucun rendez-vous",
270             customStyleSignaturePad: { border: "#c2c2c2 3px solid" },
271             appointments: [],
272             educators: [],
273         };
274     },
275     methods: {
276         /**
277          * Method to load all educators from the api rest endpoint "GET api/v1/users/
278          educators".
279          *
280          */
281         loadEducators() {
282             this.$http.get(this.$API_URL + "users/educators/").then((response) => {

```



```

282     this.educators = response.data;
283     if (this.authAdministrator && this.$route.params.userId) {
284         this.title = "Rendez-vous du client";
285         this.loadCustomerAppointmentsByUserId(this.$route.params.userId);
286     } else {
287         this.title = "Mes rendez-vous";
288         this.loadAuthCustomerAppointment();
289     }
290 });
291 },
292 /**
293  * Method to load all appointments of the authenticated client from the api
294  * rest endpoint "GET api/v1/appointments".
295  */
296 loadAuthCustomerAppointment() {
297     const config = {
298         headers: {
299             "Authorization" : this.$store.state.api_token
300         },
301     };
302     this.$http
303         .get(this.$API_URL + "appointments/", config)
304         .then((response) => {
305             this.loadAppointmentsData(response.data);
306         })
307         .catch((error) => {
308             this.$alertify.error(error.response.data.error);
309         });
310 },
311 /**
312  * Method to load all the appointments of a client from his id from the api
313  * rest endpoint "GET api/v1/users/{userId}"
314  */
315 @param {number} userId The client's id
316 loadCustomerAppointmentsByUserId(userId) {
317     const config = {
318         headers: {
319             "Authorization" : this.$store.state.api_token
320         },
321     };
322     this.$http
323         .get(this.$API_URL + "users/" + userId, config)
324         .then((response) => {
325             this.customer.firstname = response.data.firstname;
326             this.customer.lastname = response.data.lastname;
327
328             this.loadAppointmentsData(response.data.appointments);
329         })
330         .catch((error) => {
331             this.$alertify.error(error.response.data.error);
332         });
333 },
334 /**
335  * Method for formatting the time data of an appointment object table.
336  */
337 @param {Object[]} appointments The appointment object table
338 */

```

```

339 loadAppointmentsData(appointments) {
340     this.appointments = [];
341     appointments.forEach((appointment) => {
342         var date = new Date(appointment.datetime_appointment.replace(" ", "T"));
343         var options = {
344             weekday: "long",
345             year: "numeric",
346             month: "long",
347             day: "numeric",
348         };
349         var educator = this.educators.find(
350             (x) => x.id === appointment.user_id_educator
351         );
352         var educatorFullname = educator.firstname + " " + educator.lastname;
353         this.appointments.push({
354             id: appointment.id,
355             hourStart: date.getHours() + "h",
356             hourEnd: date.getHours() + appointment.duration_in_hour + "h",
357             date: date.toLocaleDateString("fr-CH", options),
358             summary: appointment.summary,
359             noteText: appointment.note_text,
360             noteGraphicalSerialId: appointment.note_graphical_serial_id,
361             educatorFullname: educatorFullname,
362         });
363     });
364 },
365 /**
366  * Method to load a graphical note from its serial id from the api rest
367  * endpoint "GET api/v1/appointments/downloadNoteGraphical/{
368  *   noteGraphicalSerialId}".
369  *
370  * @param {string} noteGraphicalSerialId The serial id of the graphical note
371  */
372 loadNoteGraphicalPicture(noteGraphicalSerialId) {
373     const config = {
374         headers: {
375             "Authorization" : this.$store.state.api_token
376         },
377         responseType: "arraybuffer",
378     };
379     this.$http
380         .get(
381             this.$API_URL +
382             "appointments/downloadNoteGraphical/" +
383             noteGraphicalSerialId,
384             config
385         )
386         .then((response) => {
387             var base64 =
388                 "data:image/png;base64," +
389                 Buffer.from(response.data, "binary").toString("base64");
390             this.$refs.signaturePad.fromDataURL(base64);
391         })
392         .catch((error) => {
393             this.$alertify.error(error.response.data.error);
394         });
395 },
396 /**
397  * Method to update an appointment from his id from the api rest endpoint "

```

```

    PATCH api/v1/appointments/{appointmentId}"
396 *
397 * @param {number} id The appointment id
398 * @param {string} summary The summary of the appointment
399 * @param {string} noteText The note text of the appointment
400 */
401 updateAppointmentByAppointmentId(id, summary, noteText) {
402     const params = new URLSearchParams();
403     params.append("summary", summary);
404     params.append("note_text", noteText);
405     const config = {
406         headers: {
407             "Authorization" : this.$store.state.api_token,
408         },
409     };
410     this.$http
411         .patch(this.$API_URL + "appointments/" + id, params, config)
412         .then((response) => {
413             console.log(response);
414             this.loadCustomerAppointmentsByUserId(this.$route.params.userId);
415             this.$alertify.success("Notes textuelles modifiées");
416             this.$bvModal.hide("modal-update-appointment-informations");
417         })
418         .catch((error) => {
419             this.$alertify.error(error.response.data.error);
420         });
421 },
422 /**
423  * Method to allow the component to read the data of an appointment in order to
424  * proceed to a modification or a deletion from a modal.
425  *
426  * @param {number} id The appointment id
427  * @param {string} summary The summary of the appointment
428  * @param {string} noteText The note text of the appointment
429  * @param {string} noteGraphicalSerialId The serial id of the graphical note of
430  * the appointment
431  */
432 sendAppointmentInformations(id, summary, noteText, noteGraphicalSerialId) {
433     this.selectedAppointment.id = id;
434     this.selectedAppointment.summary = summary;
435     this.selectedAppointment.noteText = noteText;
436     this.selectedAppointment.noteGraphicalSerialId = noteGraphicalSerialId;
437 },
438 /**
439  * Method to upload a graphical note from the api rest endpoint "POST api/v1/
440  * appointments/uploadNoteGraphical".
441  *
442  * @param {number} id The appointment id
443  * @param {string} noteGraphicalBase64 The graphical note in base64 format
444  */
445 uploadNoteGraphicalByAppointmentId(id, noteGraphicalBase64) {
446     //Workaround with the fetch api to convert a base64 to a Binary Large object
447     fetch(noteGraphicalBase64)
448         .then((res) => res.blob())
449         .then((blob) => {
450             var file = new File([blob], "note", { type: "image/jpeg" });
451             let formData = new FormData();
452             formData.append("note_graphical", file);
453             formData.append("appointment_id", id);

```

```

451     const config = {
452       headers: {
453         "Authorization" : this.$store.state.api_token
454       },
455     };
456     this.$http
457       .post(
458         this.$API_URL + "appointments/uploadNoteGraphical/",
459         formData,
460         config
461       )
462       .then((response) => {
463         console.log(response);
464         this.loadCustomerAppointmentsByUserId(this.$route.params.userId);
465         this.$alertify.success("Notes graphiques ajoutées avec succès");
466         this.$bvModal.hide("modal-graphical-note");
467       })
468       .catch((error) => {
469         this.$alertify.error(error.response.data.error);
470       });
471   });
472 },
473 /**
474  * Method to delete an appointment from his id from the api rest endpoint "
475  * DELETE api/v1/appointments/{appointmentId}"
476  *
477  * @param {number} id The appointment id
478  */
479 deleteAppointmentById(id) {
480   const config = {
481     headers: {
482       "Authorization" : this.$store.state.api_token,
483     },
484   };
485   this.$http
486     .delete(this.$API_URL + "appointments/" + id, config)
487     .then((response) => {
488       this.loadCustomerAppointmentsByUserId(this.$route.params.userId);
489       this.$alertify.success("Rendez-vous supprimé");
490       console.log(response);
491     })
492     .catch((error) => {
493       this.$alertify.error(error.response.data.error);
494     });
495 },
496 /**
497  * Method for clearing the signature of the signaturePad component
498  */
499 clearNoteGraphical() {
500   this.$refs.signaturePad.clearSignature();
501 },
502 },
503 computed: {
504   authAdministrator() {
505     return this.$store.getters.ifAdministratorAuthenticated;
506   },
507 },
508 mounted() {

```

```

509     this.loadEducators();
510   },
511 };
512 </script>
513
514 <style scoped>
515 .ComponnentContent {
516   margin-top: 20px;
517   color: #3ea3d8;
518 }
519 </style>

```

Listing 8 – ./Sources/src/components/CustomerInformation.vue

```

1  <!--
2   CustomerInformation.vue
3
4   Component representing the personal information page of a client.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7 -->
8
9 <template>
10 <div>
11   <b-container>
12     <b-row class="text-center content">
13       <b-col>
14         <h1>{{ title }}</h1>
15       </b-col>
16     </b-row>
17     <b-row class="content">
18       <b-col
19         md="4"
20         style="margin-bottom: 10px"
21         class="text-center d-flex flex-column align-items-center"
22       >
23         <b-card
24           style="padding: 10px; width: 100%; margin-bottom: 10px"
25           img-src="../../../assets/img/user-profile.png"
26           img-top
27           :title="customer.firstname"
28         >
29           <p class="text-secondary mb-1" v-if="customer.code_role == '1'">
30             Client
31           </p>
32           <p class="text-secondary mb-1">{{ customer.address }}</p>
33           <div v-if="authCustomer">
34             <b-button
35               id="btnUpdatePassword"
36               variant="outline-primary"
37               v-b-modal.modal-update-password
38               block
39             >
40               Modifier mon mot de passe
41             </b-button>
42           </div>
43           <div v-if="authAdministrator">
44             <b-button
45               id="btnGoToAppointment"

```

```

46         :to="{
47             name: 'customerAppointment',
48             params: { userId: $route.params.userId },
49         }"
50         variant="outline-primary"
51         class="btnAdmin"
52         v-if="authAdministrator"
53         >Rendez-vous
54     </b-button>
55     <b-button
56         id="btnAddDog"
57         variant="outline-primary"
58         class="btnAdmin"
59         v-if="authAdministrator"
60         v-b-modal.modal-add-dog
61     >
62         Ajouter un chien
63     </b-button>
64     <b-button
65         id="btnAddDocument"
66         variant="outline-primary"
67         class="btnAdmin"
68         v-if="authAdministrator"
69         v-b-modal.modal-add-document
70     >
71         Ajouter un document
72     </b-button>
73     <b-button
74         id="btnDeleteUser"
75         variant="outline-danger"
76         class="btnAdmin"
77         v-if="authAdministrator"
78         v-b-modal.modal-delete-user
79     >
80         Supprimer l'utilisateur
81     </b-button>
82 </div>
83 </b-card>
84
85 <b-card
86     v-if="customer.documents.length > 0"
87     style="padding: 10px; width: 100%"
88 >
89     <div v-for="document in customer.documents" :key="document.id">
90         <b-row style="text-align: center; margin-top: 10px">
91             <b-col>
92                 <b-button
93                     @click="
94                         downloadDocument(
95                             document.type,
96                             document.document_serial_id
97                         )
98                     "
99                     class="btnAdmin"
100                     block
101                     variant="outline-primary"
102                 >
103                     {{ document.type }}_{{ document.document_serial_id }}
104                     <b-icon-download></b-icon-download>

```

```

105         </b-button>
106         <b-button
107             class="btnAdmin"
108             block
109             variant="outline-danger"
110             v-if="authAdministrator"
111             v-b-modal.modal-delete-document
112             @click="sendDocumentId(document.id)"
113         >
114             Supprimer le document
115         </b-button>
116     </b-col>
117 </b-row>
118     <hr style="margin-bottom: 0px; margin-top: 0px" />
119 </div>
120 </b-card>
121 </b-col>
122 <b-col md="8">
123     <b-card style="margin-bottom: 10px">
124         <b-row>
125             <b-col sm="3">
126                 <h6 class="mb-0">Nom complet</h6>
127             </b-col>
128             <b-col sm="9" class="text-secondary">
129                 {{ customer.firstname }} {{ customer.lastname }}
130             </b-col>
131         </b-row>
132         <hr />
133         <b-row>
134             <b-col sm="3">
135                 <h6 class="mb-0">Adresse e-mail</h6>
136             </b-col>
137             <b-col sm="9" class="text-secondary">{{ customer.email }}</b-col>
138         </b-row>
139         <hr />
140         <b-row>
141             <b-col sm="3">
142                 <h6 class="mb-0">Numéro de téléphone</h6>
143             </b-col>
144             <b-col sm="9" class="text-secondary">{{
145                 customer.phonenumber
146             }}</b-col>
147         </b-row>
148         <hr />
149         <b-row>
150             <b-col sm="3">
151                 <h6 class="mb-0">Adresse de domicile</h6>
152             </b-col>
153             <b-col sm="9" class="text-secondary">{{
154                 customer.address
155             }}</b-col>
156         </b-row>
157     <div v-if="authAdministrator">
158         <hr />
159         <b-button
160             id="btnUpdateUser"
161             variant="outline-primary"
162             class="btnAdmin"
163             v-b-modal.modal-update-user

```

```

164         >
165         Modifier l'utilisateur
166     </b-button>
167 </div>
168 </b-card>
169 <b-row>
170     <b-col md="6" v-for="dog in customer.dogs" :key="dog.id">
171         <b-card style="margin-bottom: 10px">
172             <b-row no-gutters>
173                 <b-col md="12">
174                     <b-card-img
175                         v-if="dog.picture_serial_id"
176                         :src="
177                             $API_URL +
178                             'dogs/downloadPicture/' +
179                             dog.picture_serial_id
180                             "
181                         alt="Image"
182                         style="margin-bottom: 15px"
183                     >
184                     </b-card-img>
185                     <b-card-img
186                         v-else
187                         :src="require('../assets/img/placeholder-dog.png')"
188                         alt="placeholder"
189                         style="margin-bottom: 15px"
190                     >
191                     </b-card-img>
192                 </b-col>
193             </b-row>
194             <b-row>
195                 <b-col sm="4">
196                     <h6 class="mb-0">Nom</h6>
197                 </b-col>
198                 <b-col sm="8" class="text-secondary">{{ dog.name }}</b-col>
199             </b-row>
200             <hr />
201             <b-row>
202                 <b-col sm="4">
203                     <h6 class="mb-0">Race</h6>
204                 </b-col>
205                 <b-col sm="8" class="text-secondary">{{ dog.breed }}</b-col>
206             </b-row>
207             <hr />
208             <b-row>
209                 <b-col sm="4">
210                     <h6 class="mb-0">Sexe</h6>
211                 </b-col>
212                 <b-col sm="8" class="text-secondary">{{ dog.sex }}</b-col>
213             </b-row>
214             <hr />
215             <b-row>
216                 <b-col sm="4">
217                     <h6 class="mb-0">Numéro de puce</h6>
218                 </b-col>
219                 <b-col sm="8" class="text-secondary">
220                     <span v-if="dog.chip_id != ''">{{ dog.chip_id }}</span>
221                     <span v-else>-</span>
222                 </b-col>
223             </b-row>

```



```

223         <div v-if="authAdministrator">
224             <hr />
225             <b-button
226                 id="btnUpdateDog"
227                 variant="outline-primary"
228                 class="btnAdmin"
229                 v-b-modal.modal-update-dog
230                 @click="
231                     sendDogId(dog.id);
232                     sendDogInformations(
233                         dog.name,
234                         dog.breed,
235                         dog.sex,
236                         dog.chip_id
237                     );
238                 "
239             >
240                 Modifier le chien
241             </b-button>
242             <b-button
243                 id="btnAddPictureDog"
244                 class="btnAdmin"
245                 variant="outline-primary"
246                 v-b-modal.modal-add-picture-dog
247                 @click="sendDogId(dog.id)"
248             >
249                 Modifier la photo
250             </b-button>
251             <b-button
252                 id="btnDeleteDog"
253                 variant="outline-danger"
254                 class="btnAdmin"
255                 v-b-modal.modal-delete-dog
256                 @click="sendDogId(dog.id)"
257             >
258                 Supprimer le chien
259             </b-button>
260         </div>
261     </b-card>
262 </b-col>
263 </b-row>
264 </b-col>
265 </b-row>
266 <button-return
267     v-if="authAdministrator"
268     :to="{
269         name: 'administration',
270     }"
271 ></button-return>
272
273 <!-- MODAL ADD DOG-->
274 <b-modal id="modal-add-dog" title="Ajouter un chien" :hide-footer="true">
275     <b-form
276         @submit.prevent="
277             createDogForCustomerByUserId(
278                 addDogForm.name,
279                 addDogForm.breed,
280                 addDogForm.sex,
281                 addDogForm.chip_id,

```

```

282         $route.params.userId
283     )
284     "
285 >
286 <b-form-group
287     id="input-group-dog-name"
288     label="Nom :"
289     label-for="input-dog-name"
290 >
291     <b-form-input
292         id="input-dog-name"
293         v-model="addDogForm.name"
294         type="text"
295         placeholder="Entrez le nom du chien"
296         required
297     ></b-form-input>
298 </b-form-group>
299
300 <b-form-group
301     id="input-group-dog-breed"
302     label="Race :"
303     label-for="input-dog-breed"
304 >
305     <b-form-input
306         id="input-dog-breed"
307         v-model="addDogForm.breed"
308         type="text"
309         placeholder="Entrez la race du chien"
310         required
311     ></b-form-input>
312 </b-form-group>
313
314 <b-form-group
315     id="input-group-dog-sex"
316     label="Sexe :"
317     label-for="input-dog-sex"
318 >
319     <b-form-select
320         id="input-dog-sex"
321         v-model="addDogForm.sex"
322         :options="sexDogOptions"
323         required
324     ></b-form-select>
325 </b-form-group>
326
327 <b-form-group
328     id="input-group-dog-chip-id"
329     label="Numéro de puce :"
330     label-for="input-dog-chip-id"
331 >
332     <b-form-input
333         id="input-dog-chip-id"
334         v-model="addDogForm.chip_id"
335         type="text"
336         placeholder="Entrez le numéro de puce du chien"
337     ></b-form-input>
338 </b-form-group>
339
340 <b-button block type="submit" variant="outline-primary">

```

```

341         Ajouter le chien
342     </b-button>
343 </b-form>
344 </b-modal>
345
346 <!-- MODAL ADD DOG PICTURE-->
347 <b-modal
348     id="modal-add-picture-dog"
349     title="Ajouter une photo de chien"
350     :hide-footer="true"
351 >
352     <b-form
353         @submit.prevent="
354             uploadDogPictureByDogId(selectedDog.id, selectedDog.pictureFile)
355         "
356     >
357         <b-form-group
358             id="input-group-dog-picture-file"
359             label="Photo :"
360             label-for="input-dog-picture-file"
361         >
362             <b-form-file
363                 required
364                 v-model="selectedDog.pictureFile"
365                 accept="image/jpeg, image/png"
366                 placeholder="Aucun fichier choisi"
367                 browse-text="Ajouter"
368                 id="input-dog-picture-file"
369                 size="sm"
370             ></b-form-file>
371         </b-form-group>
372
373         <b-button block type="submit" variant="outline-primary">
374             Ajouter la photo
375         </b-button>
376     </b-form>
377 </b-modal>
378
379 <!-- MODAL ADD DOCUMENT-->
380 <b-modal
381     id="modal-add-document"
382     title="Ajouter un document"
383     :hide-footer="true"
384     @hidden="resetModalAddDocument"
385 >
386     <b-form
387         @submit.prevent="
388             createDocumentForCustomerByUserId(
389                 $route.params.userId,
390                 addDocumentForm.type,
391                 signatureBase64,
392                 addDocumentForm.packageNumber,
393                 addDocumentForm.file
394             )
395         "
396     >
397         <b-form-group
398             id="input-group-document-type"
399             label="Type du document : "

```

```

400     label-for="input-document-type"
401   >
402     <b-form-select
403       id="input-document-type"
404       v-model="addDocumentForm.type"
405       :options="documentOptions"
406       required
407     ></b-form-select>
408   </b-form-group>
409
410   <div v-if="addDocumentForm.type == 'conditions_inscription'">
411     <a href="/pdf/conditionsInscription.pdf" target="_blank">
412       Conditions d'inscription
413     </a>
414     <b-form-group
415       id="input-group-document-package-number"
416       label="Forfait :"
417       label-for="input-document-package-number"
418     >
419       <b-form-select
420         id="input-document-package-number"
421         v-model="addDocumentForm.packageNumber"
422         :options="packageNumbers"
423         required
424       ></b-form-select>
425     </b-form-group>
426     <signature-pad @getSignaturePad="saveSignature"></signature-pad>
427     <b-form-checkbox
428       id="checkbox-validation"
429       v-model="checkboxValidationStatus"
430       name="checkbox-validation"
431       @change="refreshStateOfcreateConditionsButton()"
432       style="margin: 10px"
433     >
434       <h5>Lu et approuvé</h5>
435     </b-form-checkbox>
436     <b-button
437       :disabled="createConditionsButtonisDisabled"
438       block
439       type="submit"
440       variant="outline-primary"
441     >
442       Créer les conditions d'inscription
443     </b-button>
444   </div>
445   <div v-else>
446     <b-form-group
447       id="input-group-document-file"
448       label="Document PDF :"
449       label-for="input-dog-document-file"
450     >
451       <b-form-file
452         required
453         v-model="addDocumentForm.file"
454         placeholder="Aucun fichier choisi"
455         browse-text="Ajouter"
456         id="input-document-file"
457         size="sm"
458       ></b-form-file>

```

```

459         </b-form-group>
460         <b-button block type="submit" variant="outline-primary">
461             Ajouter le document
462         </b-button>
463     </div>
464 </b-form>
465 </b-modal>
466
467 <!-- MODAL DELETE USER-->
468 <b-modal
469     id="modal-delete-user"
470     title="Confirmation de suppression d'utilisateur"
471     :hide-footer="true"
472     :hide-header="true"
473 >
474     <h5 style="text-align: center">Supprimer l'utilisateur ?</h5>
475     <b-form @submit.prevent="deleteUserById($route.params.userId)">
476         <b-row>
477             <b-col>
478                 <b-button
479                     block
480                     variant="danger"
481                     @click="$bvModal.hide('modal-delete-user')"
482                 >
483                     Non
484                 </b-button>
485             </b-col>
486             <b-col>
487                 <b-button block type="submit" variant="success">Oui</b-button>
488             </b-col>
489         </b-row>
490     </b-form>
491 </b-modal>
492
493 <!-- MODAL DELETE DOCUMENT-->
494 <b-modal
495     id="modal-delete-document"
496     title="Confirmation de suppression d'utilisateur"
497     :hide-footer="true"
498     :hide-header="true"
499 >
500     <h5 style="text-align: center">Supprimer le document ?</h5>
501     <b-form @submit.prevent="deleteDocumentById(selectedDocument.id)">
502         <b-row>
503             <b-col>
504                 <b-button
505                     block
506                     variant="danger"
507                     @click="$bvModal.hide('modal-delete-document')"
508                 >
509                     Non
510                 </b-button>
511             </b-col>
512             <b-col>
513                 <b-button
514                     block
515                     type="submit"
516                     variant="success"
517                     @click="$bvModal.hide('modal-delete-document')"

```

```

518         >
519         Oui
520     </b-button>
521 </b-col>
522 </b-row>
523 </b-form>
524 </b-modal>
525
526 <!-- MODAL DELETE DOG-->
527 <b-modal
528     id="modal-delete-dog"
529     title="Confirmation de suppression d'utilisateur"
530     :hide-footer="true"
531     :hide-header="true"
532 >
533     <h5 style="text-align: center">Supprimer le chien ?</h5>
534     <b-form @submit.prevent="deleteDogById(selectedDog.id)">
535         <b-row>
536             <b-col>
537                 <b-button
538                     block
539                     variant="danger"
540                     @click="$bvModal.hide('modal-delete-dog')"
541                 >
542                     Non
543                 </b-button>
544             </b-col>
545             <b-col>
546                 <b-button
547                     @click="$bvModal.hide('modal-delete-dog')"
548                     block
549                     type="submit"
550                     variant="success"
551                 >
552                     Oui
553                 </b-button>
554             </b-col>
555         </b-row>
556     </b-form>
557 </b-modal>
558
559 <!-- MODAL UPDATE USER-->
560 <b-modal
561     id="modal-update-user"
562     title="Modifier l'utilisateur"
563     :hide-footer="true"
564 >
565     <b-form
566         @submit.prevent="
567             updateUserById(
568                 $route.params.userId,
569                 customer.firstname,
570                 customer.lastname,
571                 customer.email,
572                 customer.phonenumber,
573                 customer.address
574             )
575         "
576     >

```

```

577 <b-form-group
578   id="input-group-user-firstname"
579   label="Prénom :"
580   label-for="input-user-firstname"
581 >
582   <b-form-input
583     id="input-user-firstname"
584     v-model="customer.firstname"
585     type="text"
586     placeholder="Entrez le prénom de l'utilisateur"
587     required
588   ></b-form-input>
589 </b-form-group>
590
591 <b-form-group
592   id="input-group-user-lastname"
593   label="Nom :"
594   label-for="input-user-lastname"
595 >
596   <b-form-input
597     id="input-user-lastname"
598     v-model="customer.lastname"
599     type="text"
600     placeholder="Entrez la nom de l'utilisateur"
601     required
602   ></b-form-input>
603 </b-form-group>
604
605 <b-form-group
606   id="input-group-user-email"
607   label="Adresse e-mail :"
608   label-for="input-user-email"
609 >
610   <b-form-input
611     id="input-user-email"
612     v-model="customer.email"
613     type="email"
614     placeholder="Entrez l'adresse e-mail de l'utilisateur"
615     required
616   ></b-form-input>
617 </b-form-group>
618
619 <b-form-group
620   id="input-group-user-phonenummer"
621   label="Numéro de téléphone :"
622   label-for="input-user-phonenummer"
623 >
624   <b-form-input
625     id="input-user-phonenummer"
626     v-model="customer.phonenummer"
627     type="text"
628     placeholder="Entrez le numéro de téléphone de l'utilisateur"
629     required
630   ></b-form-input>
631 </b-form-group>
632
633 <b-form-group
634   id="input-group-user-address"
635   label="Adresse de domicile :"

```

```

636         label-for="input-user-address"
637     >
638         <b-form-input
639             id="input-user-address"
640             v-model="customer.address"
641             type="text"
642             placeholder="Entrez l'adresse de domicile de l'utilisateur"
643             required
644         ></b-form-input>
645     </b-form-group>
646
647     <b-button block type="submit" variant="outline-primary">
648         Modifier l'utilisateur
649     </b-button>
650 </b-form>
651 </b-modal>
652
653 <!-- MODAL UPDATE DOG-->
654 <b-modal
655     id="modal-update-dog"
656     title="Modifier le chien"
657     :hide-footer="true"
658 >
659     <b-form
660         @submit.prevent="
661             updateDogById(
662                 selectedDog.id,
663                 selectedDog.name,
664                 selectedDog.breed,
665                 selectedDog.sex,
666                 selectedDog.chipId
667             )
668         "
669     >
670         <b-form-group
671             id="input-group-dog-name"
672             label="Nom :"
673             label-for="input-dog-name"
674         >
675             <b-form-input
676                 id="input-dog-name"
677                 v-model="selectedDog.name"
678                 type="text"
679                 placeholder="Entrez le nom du chien"
680                 required
681             ></b-form-input>
682         </b-form-group>
683
684         <b-form-group
685             id="input-group-dog-breed"
686             label="Race :"
687             label-for="input-dog-breed"
688         >
689             <b-form-input
690                 id="input-dog-breed"
691                 v-model="selectedDog.breed"
692                 type="text"
693                 placeholder="Entrez la race du chien"
694                 required

```



```

695         ></b-form-input>
696     </b-form-group>
697
698     <b-form-group
699         id="input-group-dog-sex"
700         label="Sexe :"
701         label-for="input-dog-sex"
702     >
703         <b-form-select
704             id="input-dog-sex"
705             v-model="selectedDog.sex"
706             :options="sexDogOptions"
707             required
708         ></b-form-select>
709     </b-form-group>
710
711     <b-form-group
712         id="input-group-dog-chip-id"
713         label="Numéro de puce :"
714         label-for="input-dog-chip-id"
715     >
716         <b-form-input
717             id="input-dog-chip-id"
718             v-model="selectedDog.chipId"
719             type="text"
720             placeholder="Entrez le numéro de puce du chien"
721             required
722         ></b-form-input>
723     </b-form-group>
724
725     <b-button block type="submit" variant="outline-primary">
726         Modifier le chien
727     </b-button>
728 </b-form>
729 </b-modal>
730
731 <!-- MODAL UPDATE PASSWORD-->
732 <b-modal
733     id="modal-update-password"
734     title="Modifier le mot de passe"
735     :hide-footer="true"
736 >
737     <b-form
738         @submit.prevent="updateAuthCustomerPassword(password, repeatPassword)"
739     >
740         <b-form-group
741             id="input-group-inscription-password"
742             label="Mot de passe :"
743             label-for="input-inscription-password"
744         >
745             <b-form-input
746                 id="input-inscription-password"
747                 v-model="password"
748                 type="password"
749                 placeholder="Entrez le mot de passe"
750                 required
751             ></b-form-input>
752         </b-form-group>
753     <b-form-group

```

```

754         id="input-group-inscription-second-password"
755         label="Confirmation du mot de passe :"
756         label-for="input-inscription-second-password"
757     >
758         <b-form-input
759             id="input-inscription-second-password"
760             v-model="repeatPassword"
761             type="password"
762             placeholder="Entrez le mot de passe"
763             required
764         ></b-form-input>
765     </b-form-group>
766
767     <b-button block type="submit" variant="outline-primary">
768         Modifier le mot de passe
769     </b-button>
770 </b-form>
771 </b-modal>
772 </b-container>
773 </div>
774 </template>
775
776 <script>
777 import { BIconDownload } from "bootstrap-vue";
778 import SignaturePad from "./SignaturePad.vue";
779 import ButtonReturn from "./ButtonReturn.vue";
780
781 export default {
782     components: {
783         BIconDownload,
784         SignaturePad,
785         ButtonReturn,
786     },
787     name: "CustomerInformation",
788     data() {
789         return {
790             title: null,
791             customer: {
792                 id: "",
793                 email: "",
794                 firstname: "",
795                 lastname: "",
796                 phonenumber: "",
797                 address: "",
798                 code_role: null,
799                 dogs: [],
800                 documents: [],
801             },
802             addDogForm: {
803                 name: "",
804                 breed: "",
805                 sex: "Mâle",
806                 chip_id: "",
807                 pictureFile: null,
808             },
809             selectedDog: {
810                 id: null,
811                 name: "",
812                 breed: "",

```

```

813     sex: "",
814     chipId: "",
815 },
816 sexDogOptions: ["Mâle", "Femelle"],
817 selectedDocument: {
818     id: null,
819 },
820 addDocumentForm: {
821     type: "poster",
822     packageNumber: 1,
823     file: null,
824 },
825 documentOptions: [
826     { value: "poster", text: "Document pdf" },
827     { value: "conditions_inscription", text: "Conditions d'inscription" },
828 ],
829 packageNumbers: [
830     { value: 1, text: "Bilan 'dévaluation : 70 € / 80 CHF" },
831     { value: 2, text: "Bilan + 1 séance 'déducation : 125 € / 140 CHF" },
832     { value: 3, text: "Forfait bilan + 3 séances : 230 € / 250 CHF" },
833     { value: 4, text: "Forfait bilan + 6 séances : 400 € / 440 CHF" },
834     { value: 5, text: "Forfait bilan + 9 séances : 520 € / 560 CHF" },
835 ],
836 signatureBase64: "",
837 signatureIsBlocked: false,
838 checkboxValidationStatus: false,
839 createConditionsButtonIsDisabled: true,
840 password: null,
841 repeatPassword: null,
842 };
843 },
844 methods: {
845     /**
846      * Method to load all informations of the authenticated client from the api
847      * rest endpoint "GET api/v1/users/me".
848      */
849     loadAuthCustomerInformations() {
850         const config = {
851             headers: {
852                 "Authorization" : this.$store.state.api_token
853             },
854         };
855         this.$http
856             .get(this.$API_URL + "users/me/", config)
857             .then((response) => {
858                 this.customer.id = response.data.id;
859                 this.customer.email = response.data.email;
860                 this.customer.firstname = response.data.firstname;
861                 this.customer.lastname = response.data.lastname;
862                 this.customer.phonenumber = response.data.phonenumber;
863                 this.customer.address = response.data.address;
864                 this.customer.code_role = response.data.code_role;
865                 this.customer.dogs = response.data.dogs;
866                 this.customer.documents = response.data.documents;
867             })
868             .catch((error) => {
869                 this.$alertify.error(error.response.data.error);
870             });

```

```

871 },
872 /**
873  * Method to load all informations of a client from his id from the api rest
      endpoint "GET api/v1/users/{userId}"
874  *
875  * @param {number} userId The client's id
876  */
877 loadCustomerInformationsByUserId(userId) {
878     const config = {
879         headers: {
880             "Authorization" : this.$store.state.api_token
881         },
882     };
883     this.$http
884         .get(this.$API_URL + "users/" + userId, config)
885         .then((response) => {
886             console.log(response);
887             this.customer.id = response.data.id;
888             this.customer.email = response.data.email;
889             this.customer.firstname = response.data.firstname;
890             this.customer.lastname = response.data.lastname;
891             this.customer.phonenumber = response.data.phonenumber;
892             this.customer.address = response.data.address;
893             this.customer.code_role = response.data.code_role;
894             this.customer.dogs = response.data.dogs;
895             this.customer.documents = response.data.documents;
896         })
897         .catch((error) => {
898             this.$alertify.error(error.response.data.error);
899         });
900 },
901 /**
902  * Method to create a dog for a user with the api rest endpoint "POST api/v1/
      dogs".
903  *
904  * @param {string} name The dog's name
905  * @param {string} breed The dog's breed
906  * @param {string} sex The dog's sex
907  * @param {string} chipId The dog's chip id
908  * @param {number} userId The identifier of the user owner
909  */
910 createDogForCustomerByUserId(name, breed, sex, chipId, userId) {
911     const params = new URLSearchParams();
912     params.append("name", name);
913     params.append("breed", breed);
914     params.append("sex", sex);
915     params.append("chip_id", chipId);
916     params.append("user_id", userId);
917     const config = {
918         headers: {
919             "Authorization" : this.$store.state.api_token,
920         },
921     };
922     this.$http
923         .post(this.$API_URL + "dogs/", params, config)
924         .then((response) => {
925             console.log(response);
926             this.loadCustomerInformationsByUserId(this.$route.params.userId);
927             this.$alertify.success("Chien ajouté avec succès");

```

```

928         this.$bvModal.hide("modal-add-dog");
929     })
930     .catch((error) => {
931         this.$alertify.error(error.response.data.error);
932     });
933 },
934 /**
935  * Method to upload a picture dog for a dog from the api rest endpoint "POST
    api/v1/dogs/uploadPicture".
936  *
937  * @param {number} id The dog id
938  * @param {File} pictureFile The dog picture file
939  */
940 uploadDogPictureByDogId(id, pictureFile) {
941     let formData = new FormData();
942     formData.append("dog_picture", pictureFile);
943     formData.append("dog_id", id);
944     const config = {
945         headers: {
946             "Authorization" : this.$store.state.api_token
947         },
948     };
949     this.$http
950     .post(this.$API_URL + "dogs/uploadPicture/", formData, config)
951     .then((response) => {
952         console.log(response);
953         this.loadCustomerInformationsByUserId(this.$route.params.userId);
954         this.$alertify.success("Photo du chien ajouté avec succès");
955         this.$bvModal.hide("modal-add-picture-dog");
956     })
957     .catch((error) => {
958         this.$alertify.error(error.response.data.error);
959     });
960 },
961 /**
962  * Method to create a document for a user from the api rest endpoint "POST api/
    v1/documents".
963  *
964  * @param {string} userId The user id
965  * @param {string} type The document type
966  * @param {?string} signatureBase64 The signature in base64 format for the
    conditions of registration
967  * @param {?number} packageNumber The package number for the conditions of
    registration
968  * @param {?File} file The PDF document file
969  */
970 createDocumentForCustomerByUserId(
971     userId,
972     type,
973     signatureBase64,
974     packageNumber,
975     file
976 ) {
977     var params;
978     if (type == "conditions_inscription") {
979         params = new URLSearchParams();
980         params.append("signature_base64", signatureBase64);
981         params.append("package_number", packageNumber);
982     } else {

```

```

983     params = new FormData();
984     params.append("document", file);
985 }
986 params.append("type", type);
987 params.append("user_id", userId);
988
989 const config = {
990     headers: {
991         "Authorization" : this.$store.state.api_token
992     },
993 };
994 this.$http
995     .post(this.$API_URL + "documents/", params, config)
996     .then((response) => {
997         console.log(response);
998         this.loadCustomerInformationsByUserId(this.$route.params.userId);
999         this.$alertify.success("Document ajouté avec succès");
1000         this.$bvModal.hide("modal-add-document");
1001     })
1002     .catch((error) => {
1003         this.$alertify.error(error.response.data.error);
1004     });
1005 },
1006 /**
1007  * Method to downlaod a document with the api rest endpoint "GET api/v1/
1008     documents/downloadDocument/{documentSerialId}".
1009  *
1010  * @param {string} type The document type
1011  * @param {string} serial_id The document serial id
1012  */
1012 downloadDocument(type, serial_id) {
1013     const config = {
1014         headers: {
1015             "Authorization" : this.$store.state.api_token,
1016         },
1017         responseType: "blob",
1018     };
1019     this.$http
1020         .get(this.$API_URL + "documents/downloadDocument/" + serial_id, config)
1021         .then((response) => {
1022             var blob = new Blob([response.data]);
1023             var file = window.URL.createObjectURL(blob);
1024             var a = document.createElement("a");
1025             a.href = file;
1026             a.download = type + "_" + serial_id + ".pdf";
1027             document.body.appendChild(a);
1028             a.click();
1029             document.body.removeChild(a);
1030         })
1031         .catch((error) => {
1032             this.$alertify.error(error.response.data.error);
1033         });
1034 },
1035 /**
1036  * Method to delete a user from his id from the api rest endpoint "DELETE api/
1037     v1/users/{userId}"
1038  *
1039  * @param {string} id The user id
1040  */

```

```

1040 deleteUserById(id) {
1041     const config = {
1042         headers: {
1043             "Authorization" : this.$store.state.api_token,
1044         },
1045     };
1046     this.$http
1047         .delete(this.$API_URL + "users/" + id, config)
1048         .then((response) => {
1049             this.$alertify.success("Utilisateur supprimé");
1050             this.$router.push("/administration");
1051             console.log(response);
1052         })
1053         .catch((error) => {
1054             this.$alertify.error(error.response.data.error);
1055         });
1056 },
1057 /**
1058  * Method to delete a dog from his id from the api rest endpoint "DELETE api/v1
1059  * /dogs/{dogId}"
1060  *
1061  * @param {string} id The dog id
1062  */
1062 deleteDogById(id) {
1063     const config = {
1064         headers: {
1065             "Authorization" : this.$store.state.api_token,
1066         },
1067     };
1068     this.$http
1069         .delete(this.$API_URL + "dogs/" + id, config)
1070         .then((response) => {
1071             this.loadCustomerInformationsByUserId(this.$route.params.userId);
1072             this.$alertify.success("Chien supprimé");
1073             console.log(response);
1074         })
1075         .catch((error) => {
1076             this.$alertify.error(error.response.data.error);
1077         });
1078 },
1079 /**
1080  * Method to delete a document from his id from the api rest endpoint "DELETE
1081  * api/v1/documents/{documentId}"
1082  *
1083  * @param {string} id The document id
1084  */
1084 deleteDocumentById(id) {
1085     const config = {
1086         headers: {
1087             "Authorization" : this.$store.state.api_token,
1088         },
1089     };
1090     this.$http
1091         .delete(this.$API_URL + "documents/" + id, config)
1092         .then((response) => {
1093             this.loadCustomerInformationsByUserId(this.$route.params.userId);
1094             this.$alertify.success("Document supprimé");
1095             console.log(response);
1096         })

```

```

1097         .catch((error) => {
1098             this.$alertify.error(error.response.data.error);
1099         });
1100     },
1101     /**
1102      * Method to update a user from his id from the api rest endpoint "PATCH api/v1
1103      * /users/{userId}"
1104      *
1105      * @param {string} id The user id
1106      * @param {string} firstname The user's first name
1107      * @param {string} lastname The user's last name
1108      * @param {string} email The user's email address
1109      * @param {string} phonenumber The user's phonenumber
1110      * @param {string} address The user's address
1111      */
1112     updateUserById(id, firstname, lastname, email, phonenumber, address) {
1113         const params = new URLSearchParams();
1114         params.append("firstname", firstname);
1115         params.append("lastname", lastname);
1116         params.append("email", email);
1117         params.append("phonenumber", phonenumber);
1118         params.append("address", address);
1119         const config = {
1120             headers: {
1121                 "Authorization" : this.$store.state.api_token,
1122             },
1123         };
1124         this.$http
1125             .patch(this.$API_URL + "users/" + id, params, config)
1126             .then((response) => {
1127                 console.log(response);
1128                 this.loadCustomerInformationsByUserId(this.$route.params.userId);
1129                 this.$alertify.success("Utilisateur modifié");
1130                 this.$bvModal.hide("modal-update-user");
1131             })
1132             .catch((error) => {
1133                 this.$alertify.error(error.response.data.error);
1134             });
1135     },
1136     /**
1137      * Method to update a dog from his id from the api rest endpoint "PATCH api/v1/
1138      * dogs/{dogId}"
1139      *
1140      * @param {string} id The dog id
1141      * @param {string} name The dog's name
1142      * @param {string} breed The dog's breed
1143      * @param {string} sex The dog's sex
1144      * @param {string} chipId The dog's chip id
1145      */
1146     updateDogById(id, name, breed, sex, chipId) {
1147         const params = new URLSearchParams();
1148         params.append("name", name);
1149         params.append("breed", breed);
1150         params.append("sex", sex);
1151         params.append("chip_id", chipId);
1152         const config = {
1153             headers: {
1154                 "Authorization" : this.$store.state.api_token,
1155             },

```



```

1154     };
1155     this.$http
1156       .patch(this.$API_URL + "dogs/" + id, params, config)
1157       .then((response) => {
1158         console.log(response);
1159         this.loadCustomerInformationsByUserId(this.$route.params.userId);
1160         this.$alertify.success("Chien modifié");
1161         this.$bvModal.hide("modal-update-dog");
1162       })
1163       .catch((error) => {
1164         this.$alertify.error(error.response.data.error);
1165       });
1166   },
1167   /**
1168    * Method to update the password of the authenticated user with the api rest
1169    * endpoint "PATCH api/v1/users/me/changePassword".
1170    *
1171    * @param {string} password The new password
1172    * @param {string} repeatPassword Repetition of the new password
1173    */
1174   updateAuthCustomerPassword(password, repeatPassword) {
1175     if (password !== repeatPassword) {
1176       this.$alertify.error("les mots de passes ne correspondent pas");
1177       return;
1178     }
1179     const params = new URLSearchParams();
1180     params.append("password", password);
1181     const config = {
1182       headers: {
1183         "Authorization" : this.$store.state.api_token,
1184       },
1185     };
1186     this.$http
1187       .patch(this.$API_URL + "users/me/changePassword/", params, config)
1188       .then((response) => {
1189         console.log(response);
1190         this.$alertify.success("Mot de passe modifié");
1191         this.$bvModal.hide("modal-update-password");
1192       })
1193       .catch((error) => {
1194         this.$alertify.error(error.response.data.error);
1195       });
1196   },
1197   resetModalAddDocument() {
1198     this.addDocumentForm.type = "poster";
1199     this.signatureIsBlocked = false;
1200     this.checkboxValidationStatus = false;
1201   },
1202   refreshStateOfcreateConditionsButton() {
1203     if (this.signatureIsBlocked && this.checkboxValidationStatus) {
1204       this.createConditionsButtonIsDisabled = false;
1205     } else {
1206       this.createConditionsButtonIsDisabled = true;
1207     }
1208   },
1209   /**
1210    * Method to allow the component to read the id of a dog in order to proceed to
    * a modification or a deletion from a modal.
    */

```

```

1211     * @param {number} id The dog id
1212     */
1213     sendDogId(id) {
1214         this.selectedDog.id = id;
1215     },
1216     /**
1217     * Method to allow the component to read the data of a dog in order to proceed
1218     * to a modification or a deletion from a modal.
1219     *
1220     * @param {string} name The dog's name
1221     * @param {string} breed The dog's breed
1222     * @param {string} sex The dog's sex
1223     * @param {string} chipId The dog's chip id
1224     */
1225     sendDogInformations(name, breed, sex, chipId) {
1226         this.selectedDog.name = name;
1227         this.selectedDog.breed = breed;
1228         this.selectedDog.sex = sex;
1229         this.selectedDog.chipId = chipId;
1230     },
1231     /**
1232     * Method to allow the component to read the id of a document in order to
1233     * proceed to a modification or a deletion from a modal.
1234     *
1235     * @param {number} id The document id
1236     */
1237     sendDocumentId(id) {
1238         this.selectedDocument.id = id;
1239     },
1240     saveSignature(...args) {
1241         const [base64, isBlocked] = args;
1242         this.signatureBase64 = base64;
1243         this.signatureIsBlocked = isBlocked;
1244     },
1245     watch: {
1246         signatureIsBlocked: function () {
1247             this.refreshStateOfcreateConditionsButton();
1248         },
1249     },
1250     computed: {
1251         authAdministrator() {
1252             return this.$store.getters.ifAdministratorAuthenticated;
1253         },
1254         authCustomer() {
1255             return this.$store.getters.ifCustomerAuthenticated;
1256         },
1257     },
1258     mounted() {
1259         if (this.authAdministrator && this.$route.params.userId) {
1260             this.title = "Informations du client";
1261             this.loadCustomerInformationsByUserId(this.$route.params.userId);
1262         } else {
1263             this.title = "Mes informations";
1264             this.loadAuthCustomerInformations();
1265         }
1266     },
1267 };
</script>

```

```

1268
1269 <style scoped>
1270 .content {
1271   margin-top: 20px;
1272   color: #3ea3d8;
1273 }
1274 .btnAdmin {
1275   width: 100%;
1276   margin-bottom: 10px;
1277 }
1278 </style>

```

Listing 9 – ./Sources/src/components/EducatorCalendar.vue

```

1 <!--
2   EducatorCalendar.vue
3
4   Component representing the page of the personal calendar of a dog trainer.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7 -->
8
9 <template>
10   <div>
11     <b-container>
12       <b-row id="title" class="text-center">
13         <b-col>
14           <h1>Mes rendez-vous</h1>
15         </b-col>
16       </b-row>
17       <b-row>
18         <b-col>
19           <b-button
20             id="btnCreateAppointment"
21             style="margin-bottom: 10px"
22             variant="outline-primary"
23             v-b-modal.modal-create-appointment
24             block
25           >
26             Planifier un rendez-vous
27           </b-button>
28         </b-col>
29       </b-row>
30       <b-row class="text-center">
31         <b-col>
32           <FullCalendar ref="fullCalendar" :options="calendarOptions" />
33         </b-col>
34       </b-row>
35
36       <!-- MODAL CREATE APPOINTMENT -->
37       <b-modal
38         id="modal-create-appointment"
39         title="Planifier un rendez-vous"
40         :hide-footer="true"
41       >
42         <b-form
43           @submit.prevent="
44             createAppointment(
45               formCreateAppointment.date,

```

```

46         formCreateAppointment.timestart,
47         formCreateAppointment.durationInHour,
48         formCreateAppointment.customer,
49         $store.state.user_id
50     )
51     "
52 >
53     <b-row>
54         <b-col>
55             <b-form-group label="Client :">
56                 <b-form-select
57                     id="customerSelect"
58                     v-model="formCreateAppointment.customer"
59                 >
60                     <b-form-select-option
61                         v-for="customer in customers"
62                         :key="customer.id"
63                         :value="customer.id"
64                     >{{ customer.firstname }}
65                     {{ customer.lastname }}</b-form-select-option
66                 >
67             </b-form-select>
68         </b-form-group>
69     </b-col>
70 </b-row>
71 <b-row>
72     <b-col class="d-flex justify-content-center">
73         <h5>Date</h5>
74     </b-col>
75 </b-row>
76 <b-row>
77     <b-col style="margin-bottom: 10px">
78         <b-calendar
79             block
80             id="date-calendar"
81             label-help=""
82             v-model="formCreateAppointment.date"
83         ></b-calendar>
84     </b-col>
85 </b-row>
86 <b-row>
87     <b-col>
88         <b-form-group
89             id="input-group-timeslot-timestart"
90             label="L'heure de début :"
91             label-for="input-timeslot-timestart"
92         >
93             <b-form-select
94                 id="input-timeslot-timestart"
95                 v-model="formCreateAppointment.timestart"
96                 :options="timeOptions"
97             >
98             </b-form-select>
99         </b-form-group>
100     </b-col>
101 </b-row>
102 <b-row>
103     <b-col>
104         <b-form-group

```

```

105         id="input-group-timeslot-timeend"
106         label="La durée du rendez-vous (en heure) :"
107         label-for="input-timeslot-timeend"
108     >
109         <b-form-select
110             id="input-timeslot-timeend"
111             v-model="formCreateAppointment.durationInHour"
112             :options="timeDurationOptions"
113         >
114             </b-form-select>
115         </b-form-group>
116     </b-col>
117 </b-row>
118 <b-button block type="submit" variant="outline-primary">
119     Planifier le rendez-vous
120 </b-button>
121 </b-form>
122 </b-modal>
123 </b-container>
124 </div>
125 </template>
126
127 <script>
128 import FullCalendar from "@fullcalendar/vue";
129 import dayGridPlugin from "@fullcalendar/daygrid";
130 import timeGridPlugin from "@fullcalendar/timegrid";
131 import interactionPlugin from "@fullcalendar/interaction";
132 import frLocale from "@fullcalendar/core/locales/fr";
133 import moment from "moment";
134 moment.locale("fr-ch");
135
136 export default {
137     components: {
138         FullCalendar,
139     },
140     name: "EducatorCalendar",
141     data() {
142         return {
143             calendarOptions: {
144                 plugins: [dayGridPlugin, timeGridPlugin, interactionPlugin],
145                 initialView: "dayGridMonth",
146                 headerToolbar: {
147                     left: "prev,next today",
148                     center: "title",
149                     right: "dayGridMonth,timeGridWeek,timeGridDay",
150                 },
151                 height: "auto",
152                 locale: frLocale,
153                 eventDisplay: "block",
154                 allDaySlot: false,
155                 slotMinTime: "06:00:00",
156                 slotMaxTime: "20:00:00",
157                 events: [],
158                 eventBackgroundColor: "#3ea3d8",
159                 eventBorderColor: "#e08f25",
160                 eventClick: (info) => {
161                     this.$router.push({
162                         name: "customerAppointment",
163                         params: { userId: info.event.extendedProps.customerId },

```

```

164     });
165   },
166 },
167 customers: [],
168 formCreateAppointment: {
169   customer: null,
170   date: null,
171   timestart: "07",
172   durationInHour: 2,
173 },
174 timeOptions: [
175   { value: "06", text: "06h" },
176   { value: "07", text: "07h" },
177   { value: "08", text: "08h" },
178   { value: "09", text: "09h" },
179   { value: "10", text: "10h" },
180   { value: "11", text: "11h" },
181   { value: "12", text: "12h" },
182   { value: "13", text: "13h" },
183   { value: "14", text: "14h" },
184   { value: "15", text: "15h" },
185   { value: "16", text: "16h" },
186   { value: "17", text: "17h" },
187   { value: "18", text: "18h" },
188 ],
189 timeDurationOptions: [
190   { value: 1, text: "01h" },
191   { value: 2, text: "02h" },
192   { value: 3, text: "03h" },
193   { value: 4, text: "04h" },
194 ],
195 };
196 },
197 methods: {
198   /**
199    * Method to load all the appointments of the authenticated dog educator
200    * from the api rest endpoint "GET api/v1/appointments".
201    *
202    */
203   loadEducatorAppointments() {
204     const config = {
205       headers: {
206         "Authorization" : this.$store.state.api_token
207       },
208     };
209     this.$http
210       .get(this.$API_URL + "appointments/", config)
211       .then((response) => {
212         this.calendarOptions.events = [];
213         response.data.forEach((appointment) => {
214           this.calendarOptions.events.push({
215             title: "Disponible",
216             start: moment(appointment.datetime_appointment).format(
217               "YYYY-MM-DD HH:mm:ss"
218             ),
219             end: moment(appointment.datetime_appointment)
220               .add(appointment.duration_in_hour, "hour")
221               .format("YYYY-MM-DD HH:mm:ss"),
222             extendedProps: {

```

```

223         customerId: appointment.user_id_customer,
224     },
225 });
226 });
227 })
228 .catch((error) => {
229     this.$alertify.error(error.response.data.error);
230 });
231 },
232 /**
233  * Method to load all customers from the api rest endpoint "GET api/v1/users".
234  *
235  */
236 loadCustomers() {
237     const config = {
238         headers: {
239             "Authorization" : this.$store.state.api_token
240         },
241     };
242     this.$http
243         .get(this.$API_URL + "users/", config)
244         .then((response) => {
245             this.customers = response.data;
246             this.formCreateAppointment.customer = this.customers[0].id;
247         })
248         .catch((error) => {
249             this.$alertify.error(error.response.data.error);
250         });
251 },
252 /**
253  * Method to create an appointment with the api rest endpoint "POST api/v1/
254  * appointments".
255  *
256  * @param {string} date The date of the appointment
257  * @param {string} timestart The time start of the appointment
258  * @param {number} durationInHour The duration in hour of the appointment
259  * @param {number} user_id_customer The customer's identifier
260  * @param {number} user_id_educator The dog educator's identifier
261  */
262 createAppointment(
263     date,
264     timestart,
265     durationInHour,
266     user_id_customer,
267     user_id_educator
268 ) {
269     const params = new URLSearchParams();
270     params.append("datetime_appointment", date + " " + timestart + ":00:00");
271     params.append("duration_in_hour", durationInHour);
272     params.append("user_id_customer", user_id_customer);
273     params.append("user_id_educator", user_id_educator);
274     const config = {
275         headers: {
276             "Authorization" : this.$store.state.api_token,
277         },
278     };
279     this.$http
280         .post(this.$API_URL + "appointments/", params, config)
281         .then((response) => {

```

```

281         console.log(response);
282         this.$alertify.success("Rendez-vous réservé avec succès");
283         this.$bvModal.hide("modal-create-appointment");
284         this.loadEducatorAppointments(this.selected);
285     })
286     .catch((error) => {
287         this.$alertify.error(error.response.data.error);
288     });
289 },
290 },
291 mounted() {
292     this.loadCustomers();
293     this.loadEducatorAppointments();
294 },
295 };
296 </script>
297 <style scoped>
298 #badge-free {
299     padding: 10px;
300     margin-bottom: 10px;
301     background-color: #008000;
302     border: #378d8 1px solid;
303 }
304 #title {
305     margin-top: 20px;
306     color: #3ea3d8;
307 }
308 </style>

```

Listing 10 – ./Sources/src/components/Footer.vue

```

1  <!--
2      Footer.vue
3
4      Footer component included in all application previews.
5
6      Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10     <div id="footer">
11         <b-container>
12             <b-row class="text-center">
13                 <b-col>
14                     <h1>Contact</h1>
15                 </b-col>
16             </b-row>
17             <b-row class="text-center">
18                 <b-col cols="12" sm="4">
19                     <h5>Téléphone</h5>
20                     <h6><a href="tel:+330624003368">06.24.00.33.68</a></h6>
21                 </b-col>
22                 <b-col cols="12" sm="4">
23                     <h5>Email</h5>
24                     <h6>
25                         <a href="mailto:douceurdechien@douceurdechien.com"
26                             >contact@douceurdechien.com</a>
27                     >
28                 </h6>

```



```

29     </b-col>
30     <b-col cols="12" sm="4">
31         <h5>Zone d'intervention</h5>
32         <h6>Montréal et alentours</h6>
33     </b-col>
34 </b-row>
35 </b-container>
36 <b-container fluid>
37     <b-row class="text-center">
38         <b-col style="background-color: white; color: #3ea3d8">©
39         2021 Douceur de chien -
40         <RouterLink class="link" to="/privacy_policy"
41             >politique de confidentialité</RouterLink
42         >
43         -
44         <RouterLink class="link" to="/privacy_policy"
45             >conditions d'utilisation
46         </RouterLink>
47     </b-col>
48 </b-row>
49 </b-container>
50 </div>
51 </template>
52
53 <script>
54 export default {
55     name: "Footer",
56 };
57 </script>
58
59 <style scoped>
60 a {
61     color: white;
62     text-decoration: underline;
63 }
64 #footer {
65     background-color: #3ea3d8;
66     color: white;
67     margin-top: 10px;
68 }
69 .link {
70     color: #0051ff;
71 }
72 </style>

```

Listing 11 – ./Sources/src/components/Home.vue

```

1  <!--
2    Home.vue
3
4    Component representing the home page of the application.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div>
11        <left-section-content
12            :textH1="texth1LeftSectionContent"

```

```

13     :textH3="texth3LeftSectionContent"
14     :textH5="texth5LeftSectionContent"
15     :textH6="texth6LeftSectionContent"
16     :routeBouton="routeBoutonLeftSectionContent"
17     :textButton="textButtonLeftSectionContent"
18     :imageSource="imageSourceLeftSectionContent"
19 ></left-section-content>
20 <right-section-content
21   :textH3="texth3RightSectionContent"
22   :textParagraph="textParagraphRightSectionContent"
23   :routeBouton="routeBoutonRightSectionContent"
24   :textButton="textButtonRightSectionContent"
25   :imageSource="imageSourceRightSectionContent"
26 ></right-section-content>
27 </div>
28 </template>
29
30 <script>
31 import LeftSectionContent from "../LeftSectionContent.vue";
32 import Hyron from "../assets/img/hyron.png";
33 import RightSectionContent from "../RightSectionContent.vue";
34 import BorisWithDogs from "../assets/img/borisWithDogs.png";
35
36 export default {
37   components: { LeftSectionContent, RightSectionContent },
38   name: "Home",
39   data() {
40     return {
41       texth1LeftSectionContent: "Douceur de chien",
42       texth3LeftSectionContent: "Educateur & comportementaliste canin",
43       texth5LeftSectionContent: "Vous voulez prendre rendez-vous ?",
44       texth6LeftSectionContent: "Accéder aux agendas des éducateurs canins",
45       textButtonLeftSectionContent: "Agendas",
46       imageSourceLeftSectionContent: Hyron,
47       routeBoutonLeftSectionContent: "calendar",
48
49       texth3RightSectionContent: "À propos de nous",
50       textButtonRightSectionContent: "Lire plus",
51       textParagraphRightSectionContent:
52         "Je m'appelle Boris Gourdoux, je suis le fondateur de Douceur de Chien.
53         Accompagné de Hyron, Staffie mâle et Jaya, Rhodesian Ridgeback femelle.
54         J'ai fais de l'éducation bienveillante et moderne mon leitmotiv afin d
55         'aider tous les maîtres de chiens qui feront appel à mes services. Form
56         é et diplômé en rééducation comportementale, je me ferais un plaisir de
57         vous accompagner dans l'éducation de votre chien.",
58       imageSourceRightSectionContent: BorisWithDogs,
59       routeBoutonRightSectionContent: "about",
60     };
61   },
62 };
63 </script>
64 <style scoped></style>

```

Listing 12 – ./Sources/src/components/Inscription.vue

```

1 <!--
2   Inscription.vue
3

```

```

4   Component representing the inscription page of the application.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7   -->
8
9   <template>
10    <div class="content">
11      <b-container>
12        <b-row id="title" class="text-center">
13          <b-col>
14            <h1>S'inscrire</h1>
15          </b-col>
16        </b-row>
17        <b-row class="text-center text-secondary">
18          <b-col>
19            Déjà membre ?
20            <router-link to="/connection">
21              <span role="link">Se connecter</span>
22            </router-link>
23          </b-col>
24        </b-row>
25        <b-row id="title" class="justify-content-center">
26          <b-col lg="6">
27            <b-form @submit.prevent="inscription">
28              <b-form-group
29                id="input-group-inscription-email"
30                label="Adresse e-mail :"
31                label-for="input-inscription-email"
32              >
33                <b-form-input
34                  id="input-inscription-email"
35                  v-model="form.email"
36                  type="email"
37                  placeholder="Entrez l'adresse e-mail"
38                  required
39                ></b-form-input>
40              </b-form-group>
41              <b-form-group
42                id="input-group-inscription-firstname"
43                label="Prénom :"
44                label-for="input-inscription-firstname"
45              >
46                <b-form-input
47                  id="input-inscription-firstname"
48                  v-model="form.firstname"
49                  type="text"
50                  placeholder="Entrez votre prénom"
51                  required
52                ></b-form-input>
53              </b-form-group>
54              <b-form-group
55                id="input-group-inscription-lastname"
56                label="Nom de famille :"
57                label-for="input-inscription-lastname"
58              >
59                <b-form-input
60                  id="input-inscription-lastname"
61                  v-model="form.lastname"
62                  type="text"

```

```

63         placeholder="Entrez votre nom de famille"
64         required
65     ></b-form-input>
66 </b-form-group>
67 <b-form-group
68     id="input-group-inscription-phonenummer"
69     label="Numéro de téléphone :"
70     label-for="input-inscription-phonenummer"
71 >
72     <b-form-input
73         id="input-inscription-phonenummer"
74         v-model="form.phonenummer"
75         type="text"
76         placeholder="Entrez votre numéro de téléphone"
77         required
78     ></b-form-input>
79 </b-form-group>
80 <b-form-group
81     id="input-group-inscription-address"
82     label="Adresse :"
83     label-for="input-inscription-address"
84 >
85     <b-form-input
86         id="input-inscription-address"
87         v-model="form.address"
88         type="text"
89         placeholder="Entrez votre adresse"
90         required
91     ></b-form-input>
92 </b-form-group>
93 <b-form-group
94     id="input-group-inscription-password"
95     label="Mot de passe :"
96     label-for="input-inscription-password"
97 >
98     <b-form-input
99         id="input-inscription-password"
100         v-model="form.password"
101         type="password"
102         placeholder="Entrez le mot de passe"
103         required
104     ></b-form-input>
105 </b-form-group>
106 <b-form-group
107     id="input-group-inscription-second-password"
108     label="Confirmation du mot de passe :"
109     label-for="input-inscription-second-password"
110 >
111     <b-form-input
112         id="input-inscription-second-password"
113         v-model="form.repeatPassword"
114         type="password"
115         placeholder="Entrez le mot de passe"
116         required
117     ></b-form-input>
118 </b-form-group>
119 <b-form-group
120     id="input-group-inscription-captcha"
121     label="Captcha :"

```

```

122         label-for="input-inscription-captcha"
123     >
124         <label for="robot"></label>
125         <vue-recaptcha
126             ref="recaptcha"
127             @verify="onVerify"
128             sitekey="6LfeMhEbAAAAAHuNfrr-1GCFGw8yYjkYAEb47y0w"
129             :loadRecaptchaScript="true"
130         >
131             </vue-recaptcha>
132         </b-form-group>
133         <b-button block type="submit" variant="outline-primary">
134             Inscription
135         </b-button>
136     </b-form>
137 </b-col>
138 </b-row>
139 </b-container>
140 </div>
141 </template>
142
143 <script>
144 import VueRecaptcha from "vue-recaptcha";
145
146 export default {
147     name: "Inscription",
148     components: {
149         VueRecaptcha,
150     },
151     data() {
152         return {
153             form: {
154                 email: "",
155                 firstname: "",
156                 lastname: "",
157                 phonenumber: "",
158                 address: "",
159                 password: "",
160                 repeatPassword: "",
161                 robot: false,
162                 reCAPTCHAresponse: "",
163             },
164         };
165     },
166     methods: {
167         onVerify(response) {
168             if (response) {
169                 this.form.robot = true;
170                 this.form.reCAPTCHAresponse = response;
171             }
172         },
173         /**
174          * Method to register a user with the store.
175          *
176          */
177         inscription() {
178             if (this.form.password !== this.form.repeatPassword) {
179                 this.$alertify.error("les mots de passes ne correspondent pas");
180                 return;

```

```

181     }
182
183     if (!this.form.robot) {
184         this.$alertify.error("la vérification du captcha a échoué");
185         return;
186     }
187
188     this.$store.dispatch("inscription", {
189         email: this.form.email,
190         firstname: this.form.firstname,
191         lastname: this.form.lastname,
192         phonenumber: this.form.phonenumber,
193         address: this.form.address,
194         password: this.form.password,
195         reCAPTCHAuserResponseToken: this.form.reCAPTCHAresponse,
196     });
197     },
198 },
199 };
200 </script>
201
202 <style scoped>
203 #title {
204     margin-top: 20px;
205     color: #3ea3d8;
206 }
207 </style>

```

Listing 13 – ./Sources/src/components/LeftSectionContent.vue

```

1 <!--
2 LeftSectionContent.vue
3
4 Reusable component representing an aesthetic content displayed with textual
5 content on the left and an image on the right.
6
7 Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
8 -->
9 <template>
10 <div class="content">
11 <b-container>
12 <b-row>
13 <b-col class="text-content" cols="12" sm="12" lg="6">
14 <h1 class="font-weight-bold display-4">{{ textH1 }}</h1>
15 <h3 class="font-weight-bold">{{ textH3 }}</h3>
16 <h5>{{ textH5 }}</h5>
17 <h6>{{ textH6 }}</h6>
18 <p class="text-justify">{{ textParagraph }}</p>
19 <b-button id="button" :to="routeBouton" pill v-if="textButton != ''">
20 {{ textButton }}</b-button>
21 >
22 </b-col>
23 <b-col
24 class="d-flex align-items-center justify-content-center"
25 cols="12"
26 sm="12"
27 lg="6"
28 >

```

```

29         <b-img
30             v-if="imageSource != null"
31             class="circleImage"
32             :src="imageSource"
33             fluid
34             rounded="circle"
35             alt="Circle image"
36         ></b-img>
37     </b-col>
38 </b-row>
39 </b-container>
40 </div>
41 </template>
42
43 <script>
44 export default {
45     props: {
46         textParagraph: {
47             type: String,
48             default: "",
49         },
50         textH1: {
51             type: String,
52             default: "",
53         },
54         textH3: {
55             type: String,
56             default: "",
57         },
58         textH5: {
59             type: String,
60             default: "",
61         },
62         textH6: {
63             type: String,
64             default: "",
65         },
66         routeBouton: {
67             type: String,
68             default: "",
69         },
70         textButton: {
71             type: String,
72             default: "",
73         },
74         imageSource: {
75             type: String,
76             default: "",
77         },
78     },
79     name: "LeftSectionContent",
80 };
81 </script>
82
83 <style scoped>
84 .content {
85     background-color: #fef6ec;
86     padding-top: 20px;
87     padding-bottom: 20px;

```

```

88     color: #3ea3d8;
89 }
90 #button {
91     background-color: #3ea3d8;
92 }
93 </style>

```

Listing 14 – ./Sources/src/components/Navbar.vue

```

1  <!--
2      Navbar.vue
3
4      Navbar component included in all application previews.
5
6      Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10     <div>
11         <b-navbar fixed="top" toggleable="lg" type="dark" variant="dark">
12             <b-img
13                 src="../assets/img/logoblancblanc.png"
14                 v-bind="logo"
15                 fluid
16                 @click="$router.push('/') "
17                 alt="Responsive image"
18             ></b-img>
19             <b-navbar-toggle target="nav-collapse"></b-navbar-toggle>
20             <b-collapse id="nav-collapse" is-nav>
21                 <b-navbar-nav>
22                     <b-nav-item to="/" href="#">Accueil</b-nav-item>
23                     <b-nav-item to="/about" href="#">À propos</b-nav-item>
24                     <b-nav-item to="/calendar" href="#">Agenda</b-nav-item>
25                     <b-nav-item to="/customer_information" v-if="authCustomer" href="#">
26                         Mes informations
27                     </b-nav-item>
28                     <b-nav-item to="/customer_appointment" v-if="authCustomer" href="#">
29                         Mes rendez-vous
30                     </b-nav-item>
31                     <b-nav-item to="/administration" v-if="authAdministrator" href="#">
32                         Administration
33                     </b-nav-item>
34                     <b-nav-item to="/planning" v-if="authAdministrator" href="#">
35                         Mon planning
36                     </b-nav-item>
37                     <b-nav-item to="/educator_calendar" v-if="authAdministrator" href="#">
38                         Mes rendez-vous
39                     </b-nav-item>
40                 </b-navbar-nav>
41                 <b-navbar-nav class="ml-auto">
42                     <b-nav-item v-if="auth" @click="logout()" href="#">
43                         Déconnexion
44                     </b-nav-item>
45                     <b-nav-item v-if="!auth" to="/connection" href="#">
46                         Connexion
47                     </b-nav-item>
48                 </b-navbar-nav>
49             </b-collapse>
50         </b-navbar>

```



```

51     </div>
52 </template>
53
54 <script>
55 export default {
56   name: "Navbar",
57   computed: {
58     auth() {
59       return this.$store.getters.isAuthenticated;
60     },
61     authCustomer() {
62       return this.$store.getters.ifCustomerAuthenticated;
63     },
64     authAdministrator() {
65       return this.$store.getters.ifAdministratorAuthenticated;
66     },
67   },
68   methods: {
69     logout() {
70       this.$store.dispatch("logout");
71     },
72   },
73   data() {
74     return {
75       logo: { width: 80 },
76     };
77   },
78 };
79 </script>
80
81 <style scoped>
82 .navbar.navbar-dark.bg-dark {
83   background-color: #3ea3d8 !important;
84   font-size: 20px;
85 }
86 .navbar-dark .navbar-nav .nav-link {
87   color: white;
88   -webkit-transition: color 0.2s ease-in;
89   -moz-transition: color 0.2s ease-in;
90   -o-transition: color 0.2s ease-in;
91   transition: color 0.2s ease-in;
92   text-align: center;
93 }
94 .navbar-dark .navbar-nav .router-link-exact-active {
95   color: #e08f25;
96   font-weight: 500;
97 }
98 .navbar-dark .navbar-nav .nav-link:hover {
99   color: #e08f25;
100   font-weight: 500;
101 }
102 .navbar-collapse .nav-item .nav-link {
103   margin: auto;
104   font-weight: 500;
105 }
106 </style>

```

Listing 15 – ./Sources/src/components/Planning.vue

```

1  <!--
2  Planning.vue
3
4  Component representing the planning page of a dog trainer.
5
6  Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10   <div>
11     <b-container>
12       <b-row>
13         <b-col>
14           <h1 class="title text-center">Mon planning</h1>
15         </b-col>
16       </b-row>
17       <b-row>
18         <b-col>
19           <h3 class="title text-center">Calendriers hebdomadaires</h3>
20         </b-col>
21       </b-row>
22       <b-row>
23         <b-table
24           table-variant="primary"
25           striped
26           hover
27           :items="weeklySchedules"
28           :fields="tblWeeklySchedulesFields"
29         >
30           <template #cell(créneaux_horaires)="row">
31             <b-button
32               size="sm"
33               @click="row.toggleDetails"
34               class="mr-2 tableBtn"
35             >
36               {{ row.detailsShowing ? "Cacher" : "Afficher" }}
37             </b-button>
38           </template>
39           <template #row-details="row">
40             <b-table
41               table-variant="light"
42               striped
43               hover
44               :items="row.item.timeSlots"
45               :fields="tblTimeSlotsWeeklySchedulesFields"
46             >
47               <template #cell(supprimer)="row">
48                 <b-button
49                   v-b-modal.modal-delete-timeslot
50                   @click="sendTimeSlotId(row.item.id)"
51                   variant="danger"
52                 >
53                   <b-icon-trash></b-icon-trash>
54                 </b-button>
55               </template>
56             </b-table>
57             <b-button
58               variant="primary"
59               class="bg-light text-primary"

```

```

60         v-b-modal.modal-add-timeslot-weeklyschedule
61         block
62         @click="sendWeeklyScheduleId(row.item.id)"
63     >
64         Ajouter un créneau horaire
65     </b-button>
66 </template>
67 <template #cell(supprimer)="row">
68     <b-button
69         v-b-modal.modal-delete-weeklyschedule
70         @click="sendWeeklyScheduleId(row.item.id)"
71         variant="danger"
72     >
73         <b-icon-trash></b-icon-trash>
74     </b-button>
75 </template>
76 </b-table>
77 </b-row>
78 <b-row>
79     <b-col>
80         <b-button
81             id="btnAddWeeklySchedule"
82             v-b-modal.modal-add-weeklyschedule
83             variant="primary"
84             block
85         >
86             Ajouter un calendrier hebdomadaire
87         </b-button>
88     </b-col>
89 </b-row>
90 <b-row>
91     <b-col>
92         <h3 class="title text-center">Exceptions d'horaire</h3>
93     </b-col>
94 </b-row>
95 <b-row>
96     <b-table
97         table-variant="primary"
98         striped
99         hover
100         :items="scheduleOverrides"
101         :fields="tblScheduleOverridesFields"
102     >
103         <template #cell(créneaux_horaires)="row">
104             <b-button
105                 size="sm"
106                 @click="row.toggleDetails"
107                 class="mr-2 tableBtn"
108             >
109                 {{ row.detailsShowing ? "Cacher" : "Afficher" }}
110             </b-button>
111         </template>
112         <template #row-details="row">
113             <b-table
114                 table-variant="light"
115                 striped
116                 hover
117                 :items="row.item.timeSlots"
118                 :fields="tblTimeSlotsScheduleOverridesFields"

```

```

119         >
120         <template #cell(supprimer)="row">
121             <b-button
122                 v-b-modal.modal-delete-timeslot
123                 @click="sendTimeSlotId(row.item.id)"
124                 variant="danger"
125             >
126                 <b-icon-trash></b-icon-trash>
127             </b-button>
128         </template>
129     </b-table>
130     <b-button
131         variant="primary"
132         class="bg-light text-primary"
133         v-b-modal.modal-add-timeslot-scheduleoverride
134         block
135         @click="sendScheduleOverrideId(row.item.id)"
136     >
137         Ajouter un créneau horaire
138     </b-button>
139 </template>
140 <template #cell(supprimer)="row">
141     <b-button
142         v-b-modal.modal-delete-scheduleoverride
143         @click="sendScheduleOverrideId(row.item.id)"
144         variant="danger"
145     >
146         <b-icon-trash></b-icon-trash>
147     </b-button>
148 </template>
149 </b-table>
150 </b-row>
151 <b-row>
152     <b-col>
153         <b-button
154             id="btnAddScheduleOverride"
155             v-b-modal.modal-add-scheduleoverride
156             variant="primary"
157             block
158         >
159             Ajouter une exception d'horaire
160         </b-button>
161     </b-col>
162 </b-row>
163
164 <b-row>
165     <b-col>
166         <h3 class="title text-center">Vacances</h3>
167     </b-col>
168 </b-row>
169 <b-row>
170     <b-table
171         table-variant="primary"
172         striped
173         hover
174         :items="absences"
175         :fields="tblAbsencesFields"
176     >
177         <template #cell(supprimer)="row">

```

```

178         <b-button
179             v-b-modal.modal-delete-absence
180             @click="sendAbsenceId(row.item.id)"
181             variant="danger"
182         >
183             <b-icon-trash></b-icon-trash>
184         </b-button>
185     </template>
186 </b-table>
187 </b-row>
188 <b-row>
189     <b-col>
190         <b-button
191             id="btnAddAbsence"
192             v-b-modal.modal-add-absence
193             variant="primary"
194             block
195         >
196             Ajouter des vacances
197         </b-button>
198     </b-col>
199 </b-row>
200
201 <!-- MODAL ADD WEEKLYSCHEDULE -->
202 <b-modal
203     id="modal-add-weeklyschedule"
204     title="Ajouter un calendrier hebdomadaire"
205     :hide-footer="true"
206 >
207     <b-form
208         @submit.prevent="
209             createWeeklySchedule(
210                 formWeeklySchedule.date_valid_from,
211                 formWeeklySchedule.date_valid_to
212             )
213         "
214     >
215         <b-form-group
216             id="input-group-weeklyschedule-datevalidfrom"
217             label="Date de début :"
218             label-for="input-weeklyschedule-datevalidfrom"
219         >
220             <b-form-datepicker
221                 id="input-weeklyschedule-datevalidfrom"
222                 v-model="formWeeklySchedule.date_valid_from"
223                 menu-class="w-100"
224                 label-no-date-selected="Aucune date sélectionnée"
225                 calendar-width="100%"
226                 label-help=""
227                 class="mb-2"
228             >
229             </b-form-datepicker>
230         </b-form-group>
231         <b-form-group
232             id="input-group-weeklyschedule-datevalidto"
233             label="Date de fin :"
234             label-for="input-weeklyschedule-datevalidto"
235         >
236             <b-form-datepicker

```

```

237         id="input-weeklyschedule-datevalidto"
238         v-model="formWeeklySchedule.date_valid_to"
239         menu-class="w-100"
240         label-no-date-selected="Aucune date sélectionnée"
241         calendar-width="100%"
242         label-help=""
243         class="mb-2"
244     >
245 </b-form-datepicker>
246 <b-button
247     size="sm"
248     variant="outline-danger"
249     v-if="formWeeklySchedule.date_valid_to"
250     @click="clearWeeklyScheduleDateValidTo"
251 >
252     Effacer la date
253 </b-button>
254 </b-form-group>
255
256 <b-button block type="submit" variant="outline-primary">
257     Ajouter le calendrier hebdomadaire
258 </b-button>
259 </b-form>
260 </b-modal>
261
262 <!-- MODAL ADD SCHEDULEOVERRIDE -->
263 <b-modal
264     id="modal-add-scheduleoverride"
265     title="Ajouter un exception d'horaire"
266     :hide-footer="true"
267 >
268     <b-form
269         @submit.prevent="
270             createScheduleOverride(formScheduleOverride.date_schedule_override)
271         "
272     >
273         <b-form-group
274             id="input-group-scheduleoverride-datescheduleoverride"
275             label="Date de début :"
276             label-for="input-scheduleoverride-datescheduleoverride"
277         >
278             <b-form-datepicker
279                 id="input-scheduleoverride-datescheduleoverride"
280                 v-model="formScheduleOverride.date_schedule_override"
281                 menu-class="w-100"
282                 label-no-date-selected="Aucune date sélectionnée"
283                 calendar-width="100%"
284                 label-help=""
285                 class="mb-2"
286             >
287             </b-form-datepicker>
288         </b-form-group>
289
290         <b-button block type="submit" variant="outline-primary">
291             Ajouter l'exception d'horaire
292         </b-button>
293     </b-form>
294 </b-modal>
295

```

```

296 <!-- MODAL ADD ABSENCE -->
297 <b-modal
298   id="modal-add-absence"
299   title="Ajouter des vacances"
300   :hide-footer="true"
301 >
302   <b-form
303     @submit.prevent="
304       createAbsence(
305         formAbsence.date_absence_from,
306         formAbsence.date_absence_to
307       )
308   "
309 >
310   <b-form-group
311     id="input-group-absence-dateabsencefrom"
312     label="Date de début :"
313     label-for="input-absence-dateabsencefrom"
314   >
315     <b-form-datepicker
316       id="input-absence-dateabsencefrom"
317       v-model="formAbsence.date_absence_from"
318       menu-class="w-100"
319       label-no-date-selected="Aucune date sélectionnée"
320       calendar-width="100%"
321       label-help=""
322       class="mb-2"
323     >
324     </b-form-datepicker>
325   </b-form-group>
326
327   <b-form-group
328     id="input-group-absence-dateabsenceto"
329     label="Date de début :"
330     label-for="input-absence-dateabsenceto"
331   >
332     <b-form-datepicker
333       id="input-absence-dateabsenceto"
334       v-model="formAbsence.date_absence_to"
335       menu-class="w-100"
336       label-no-date-selected="Aucune date sélectionnée"
337       calendar-width="100%"
338       class="mb-2"
339     >
340     </b-form-datepicker>
341   </b-form-group>
342
343   <b-button block type="submit" variant="outline-primary">
344     Ajouter les vacances
345   </b-button>
346 </b-form>
347 </b-modal>
348
349 <!-- MODAL ADD TIMESLOT FOR WEEKLYSCHEDULE -->
350 <b-modal
351   id="modal-add-timeslot-weeklyschedule"
352   title="Ajouter un créneau horaire"
353   :hide-footer="true"
354 >

```

```

355 <b-form
356   @submit.prevent="
357     createTimeSlotForWeeklySchedule(
358       selectedWeeklyScheduleId,
359       formTimeSlot.code_day,
360       formTimeSlot.time_start,
361       formTimeSlot.time_end
362     )
363   "
364 >
365   <b-form-group
366     id="input-group-timeslot-codeday"
367     label="Jour de la semaine :"
368     label-for="input-timeslot-codeday"
369   >
370     <b-form-select
371       id="input-timeslot-codeday"
372       v-model="formTimeSlot.code_day"
373       :options="dayOptions"
374     >
375     </b-form-select>
376   </b-form-group>
377   <b-form-group
378     id="input-group-timeslot-timestart"
379     label="L'heure de début :"
380     label-for="input-timeslot-timestart"
381   >
382     <b-form-select
383       id="input-timeslot-timestart"
384       v-model="formTimeSlot.time_start"
385       :options="timeOptions"
386     >
387     </b-form-select>
388   </b-form-group>
389   <b-form-group
390     id="input-group-timeslot-timeend"
391     label="L'heure de fin :"
392     label-for="input-timeslot-timeend"
393   >
394     <b-form-select
395       id="input-timeslot-timeend"
396       v-model="formTimeSlot.time_end"
397       :options="timeOptions"
398     >
399     </b-form-select>
400   </b-form-group>
401
402   <b-button block type="submit" variant="outline-primary">
403     Ajouter le créneau horaire
404   </b-button>
405 </b-form>
406 </b-modal>
407
408 <!-- MODAL ADD TIMESLOT FOR SCHEDULE OVERRIDE -->
409 <b-modal
410   id="modal-add-timeslot-scheduleoverride"
411   title="Ajouter un créneau horaire"
412   :hide-footer="true"
413 >

```



```

414 <b-form
415   @submit.prevent="
416     createTimeSlotForScheduleOverride(
417       selectedScheduleOverrideId,
418       formTimeSlot.time_start,
419       formTimeSlot.time_end
420     )
421   "
422 >
423   <b-form-group
424     id="input-group-timeslot-timestart"
425     label="L'heure de début :"
426     label-for="input-timeslot-timestart"
427   >
428     <b-form-select
429       id="input-timeslot-timestart"
430       v-model="formTimeSlot.time_start"
431       :options="timeOptions"
432     >
433     </b-form-select>
434   </b-form-group>
435   <b-form-group
436     id="input-group-timeslot-timeend"
437     label="L'heure de fin :"
438     label-for="input-timeslot-timeend"
439   >
440     <b-form-select
441       id="input-timeslot-timeend"
442       v-model="formTimeSlot.time_end"
443       :options="timeOptions"
444     >
445     </b-form-select>
446   </b-form-group>
447
448   <b-button block type="submit" variant="outline-primary">
449     Ajouter le créneau horaire
450   </b-button>
451 </b-form>
452 </b-modal>
453
454 <!-- MODAL DELETE WEEKLY SCHEDULE-->
455 <b-modal
456   id="modal-delete-weeklyschedule"
457   title="Confirmation de suppression d'un calendrier hebdomadaire"
458   :hide-footer="true"
459   :hide-header="true"
460 >
461   <h5 style="text-align: center">
462     Supprimer le calendrier hebdomadaire ?
463   </h5>
464   <b-form
465     @submit.prevent="deleteWeeklyScheduleById(selectedWeeklyScheduleId)"
466   >
467     <b-row>
468       <b-col>
469         <b-button
470           block
471           variant="danger"
472           @click="$bvModal.hide('modal-delete-weeklyschedule')"

```

```

473         >
474         Non
475     </b-button>
476 </b-col>
477 <b-col>
478     <b-button
479         @click="$bvModal.hide('modal-delete-weeklyschedule')"
480         block
481         type="submit"
482         variant="success"
483     >
484         Oui
485     </b-button>
486 </b-col>
487 </b-row>
488 </b-form>
489 </b-modal>
490
491 <!-- MODAL DELETE SCHEDULE OVERRIDE-->
492 <b-modal
493     id="modal-delete-scheduleoverride"
494     title="Confirmation de suppression d'une exception d'horaire"
495     :hide-footer="true"
496     :hide-header="true"
497 >
498     <h5 style="text-align: center">Supprimer l'exception d'horaire ?</h5>
499     <b-form
500         @submit.prevent="
501             deleteScheduleOverrideById(selectedScheduleOverrideId)
502         "
503     >
504         <b-row>
505             <b-col>
506                 <b-button
507                     block
508                     variant="danger"
509                     @click="$bvModal.hide('modal-delete-scheduleoverride')"
510                 >
511                     Non
512                 </b-button>
513             </b-col>
514             <b-col>
515                 <b-button
516                     @click="$bvModal.hide('modal-delete-scheduleoverride')"
517                     block
518                     type="submit"
519                     variant="success"
520                 >
521                     Oui
522                 </b-button>
523             </b-col>
524         </b-row>
525     </b-form>
526 </b-modal>
527
528 <!-- MODAL DELETE ABSENCE-->
529 <b-modal
530     id="modal-delete-absence"
531     title="Confirmation de suppression des vacances"

```

```

532     :hide-footer="true"
533     :hide-header="true"
534 >
535     <h5 style="text-align: center">Supprimer les vacances ?</h5>
536     <b-form @submit.prevent="deleteAbsenceById(selectedAbsenceId)">
537         <b-row>
538             <b-col>
539                 <b-button
540                     block
541                     variant="danger"
542                     @click="$bvModal.hide('modal-delete-absence') "
543                 >
544                     Non
545                 </b-button>
546             </b-col>
547             <b-col>
548                 <b-button
549                     @click="$bvModal.hide('modal-delete-absence') "
550                     block
551                     type="submit"
552                     variant="success"
553                 >
554                     Oui
555                 </b-button>
556             </b-col>
557         </b-row>
558     </b-form>
559 </b-modal>
560
561 <!-- MODAL DELETE TIME SLOT -->
562 <b-modal
563     id="modal-delete-timeslot"
564     title="Confirmation de suppression d'un créneau horaire"
565     :hide-footer="true"
566     :hide-header="true"
567 >
568     <h5 style="text-align: center">Supprimer le créneau horaire ?</h5>
569     <b-form @submit.prevent="deleteTimeSlotById(selectedTimeSlotId)">
570         <b-row>
571             <b-col>
572                 <b-button
573                     block
574                     variant="danger"
575                     @click="$bvModal.hide('modal-delete-timeslot') "
576                 >
577                     Non
578                 </b-button>
579             </b-col>
580             <b-col>
581                 <b-button
582                     @click="$bvModal.hide('modal-delete-timeslot') "
583                     block
584                     type="submit"
585                     variant="success"
586                 >
587                     Oui
588                 </b-button>
589             </b-col>
590         </b-row>

```

```

591         </b-form>
592     </b-modal>
593 </b-container>
594 </div>
595 </template>
596
597 <script>
598 import { BIconTrash } from "bootstrap-vue";
599 import moment from "moment";
600 moment.locale("fr-ch");
601
602 export default {
603     components: {
604         BIconTrash,
605     },
606     name: "Planning",
607     data() {
608         return {
609             weeklySchedules: [],
610             timeSlots: [],
611             scheduleOverrides: [],
612             absences: [],
613             selectedWeeklyScheduleId: null,
614             selectedScheduleOverrideId: null,
615             selectedTimeSlotId: null,
616             selectedAbsenceId: null,
617             formTimeSlot: {
618                 code_day: 2,
619                 time_start: "07",
620                 time_end: "09",
621             },
622             formWeeklySchedule: {
623                 date_valid_from: null,
624                 date_valid_to: null,
625             },
626             formScheduleOverride: {
627                 date_schedule_override: null,
628             },
629             formAbsence: {
630                 date_absence_from: null,
631                 date_absence_to: null,
632             },
633             dayOptions: [
634                 { value: 2, text: "Lundi" },
635                 { value: 3, text: "Mardi" },
636                 { value: 4, text: "Mercredi" },
637                 { value: 5, text: "Jeudi" },
638                 { value: 6, text: "Vendredi" },
639                 { value: 7, text: "Samedi" },
640                 { value: 1, text: "Dimanche" },
641             ],
642             timeOptions: [
643                 { value: "06", text: "06h" },
644                 { value: "07", text: "07h" },
645                 { value: "08", text: "08h" },
646                 { value: "09", text: "09h" },
647                 { value: "10", text: "10h" },
648                 { value: "11", text: "11h" },
649                 { value: "12", text: "12h" },

```

```

650     { value: "13", text: "13h" },
651     { value: "14", text: "14h" },
652     { value: "15", text: "15h" },
653     { value: "16", text: "16h" },
654     { value: "17", text: "17h" },
655     { value: "18", text: "18h" },
656 ],
657 dayOfWeek: [
658     "",
659     "Dimanche",
660     "Lundi",
661     "Mardi",
662     "Mercredi",
663     "Jeudi",
664     "Vendredi",
665     "Samedi",
666 ],
667 tblWeeklySchedulesFields: [
668     "date_de_début",
669     "date_de_fin",
670     "créneaux horaires",
671     "supprimer",
672 ],
673 tblScheduleOverridesFields: ["date", "créneaux horaires", "supprimer"],
674 tblAbsencesFields: ["date_de_début", "date_de_fin", "supprimer"],
675 tblTimeSlotsWeeklySchedulesFields: [
676     "jour",
677     "heure_de_début",
678     "heure_de_fin",
679     "supprimer",
680 ],
681 tblTimeSlotsScheduleOverridesFields: [
682     "heure_de_début",
683     "heure_de_fin",
684     "supprimer",
685 ],
686 };
687 },
688 methods: {
689     /**
690      * Method to load and format all the weekly schedules with their time
691      * slots of the authenticated dog educator from the api rest endpoint "GET api/
692      * v1/weeklySchedules".
693      */
694     loadEducatorWeeklySchedulesWithTimeSlots() {
695         const config = {
696             headers: {
697                 "Authorization" : this.$store.state.api_token,
698             },
699         };
700         this.$http
701             .get(this.$API_URL + "weeklySchedules/", config)
702             .then((response) => {
703                 this.weeklySchedules = [];
704                 response.data.forEach((weeklySchedule) => {
705                     weeklySchedule["date_de_début"] = this.formatDate(
706                         weeklySchedule["date_valid_from"]
707                     );

```

```

708     delete weeklySchedule["date_valid_from"];
709     weeklySchedule["date_de_fin"] =
710         weeklySchedule["date_valid_to"] == undefined
711             ? "-"
712             : this.formatDate(weeklySchedule["date_valid_to"]);
713     delete weeklySchedule["date_valid_to"];
714
715     weeklySchedule["timeSlots"].forEach((timeslot) => {
716         timeslot["jour"] = this.dayOfWeek(timeslot["code_day"]);
717         delete timeslot["code_day"];
718         timeslot["heure_de_début"] = this.formatTime(
719             timeslot["time_start"]
720         );
721         delete timeslot["time_start"];
722         timeslot["heure_de_fin"] = this.formatTime(timeslot["time_end"]);
723         delete timeslot["time_end"];
724     });
725 });
726 this.weeklySchedules = response.data;
727 })
728 .catch((error) => {
729     this.$alertify.error(error.response.data.error);
730 });
731 },
732 /**
733  * Method to load and format all the schedule overrides with their time slots
734  * of the authenticated dog educator from the api rest endpoint "GET api/v1/
735  * scheduleOverrides".
736  */
737 loadEducatorScheduleOverridesWithTimeSlots() {
738     const config = {
739         headers: {
740             // eslint-disable-next-line prettier/prettier
741             "Authorization" : this.$store.state.api_token,
742         },
743     };
744     this.$http
745         .get(this.$API_URL + "scheduleOverrides/", config)
746         .then((response) => {
747             this.scheduleOverrides = [];
748             response.data.forEach((scheduleOverride) => {
749                 scheduleOverride["date"] = this.formatDate(
750                     scheduleOverride["date_schedule_override"]
751                 );
752                 delete scheduleOverride["date_schedule_override"];
753
754                 scheduleOverride["timeSlots"].forEach((timeslot) => {
755                     timeslot["heure_de_début"] = this.formatTime(
756                         timeslot["time_start"]
757                     );
758                     delete timeslot["time_start"];
759                     timeslot["heure_de_fin"] = this.formatTime(timeslot["time_end"]);
760                     delete timeslot["time_end"];
761                 });
762             });
763             this.scheduleOverrides = response.data;
764         })
765         .catch((error) => {

```

```

766         this.$alertify.error(error.response.data.error);
767     });
768 },
769 /**
770  * Method to load and format all absences of the authenticated dog educator
771  * from the api rest endpoint "GET api/v1/scheduleOverrides".
772  */
773 loadEducatorAbsences() {
774     const config = {
775         headers: {
776             // eslint-disable-next-line prettier/prettier
777             "Authorization" : this.$store.state.api_token,
778         },
779     };
780     this.$http
781     .get(this.$API_URL + "absences/", config)
782     .then((response) => {
783         this.absences = [];
784         response.data.forEach((absence) => {
785             absence["date_de_début"] = this.formatDate(
786                 absence["date_absence_from"]
787             );
788             delete absence["date_absence_from"];
789             absence["date_de_fin"] = this.formatDate(
790                 absence["date_absence_to"]
791             );
792             delete absence["date_absence_to"];
793         });
794         this.absences = response.data;
795     })
796     .catch((error) => {
797         this.$alertify.error(error.response.data.error);
798     });
799 },
800 /**
801  * Method to create a time slot for a weekly schedule from the api rest
802  * endpoint "POST api/v1/timeSlots".
803  *
804  * @param {string} weeklyScheduleId The weekly schedule id
805  * @param {string} codeDay The code day of time slot
806  * @param {string} timeStart The time start of time slot
807  * @param {?number} timeEnd The time end of time slot
808  */
809 createTimeSlotForWeeklySchedule(
810     weeklyScheduleId,
811     codeDay,
812     timeStart,
813     timeEnd
814 ) {
815     const params = new URLSearchParams();
816     params.append("id_weekly_schedule", weeklyScheduleId);
817     params.append("code_day", codeDay);
818     params.append("time_start", timeStart + ":00:00");
819     params.append("time_end", timeEnd + ":00:00");
820     const config = {
821         headers: {
822             "Authorization" : this.$store.state.api_token,

```

```

823     };
824     this.$http
825       .post(this.$API_URL + "timeSlots/", params, config)
826       .then((response) => {
827         console.log(response);
828         this.loadEducatorWeeklySchedulesWithTimeSlots();
829         this.$alertify.success("Créneau horaire ajouté avec succès");
830       })
831       .catch((error) => {
832         this.$alertify.error(error.response.data.error);
833       });
834   },
835   /**
836    * Method to create a time slot for a schedule override from the api rest
837    * endpoint "POST api/v1/timeSlots".
838    *
839    * @param {string} weeklyScheduleId The weekly schedule id
840    * @param {string} codeDay The code day of time slot
841    * @param {string} timeStart The time start of time slot
842    * @param {?number} timeEnd The time end of time slot
843    */
844   createTimeSlotForScheduleOverride(scheduleOverrideId, timeStart, timeEnd) {
845     const params = new URLSearchParams();
846     params.append("id_schedule_override", scheduleOverrideId);
847     params.append("code_day", 1);
848     params.append("time_start", timeStart + ":00:00");
849     params.append("time_end", timeEnd + ":00:00");
850     const config = {
851       headers: {
852         "Authorization" : this.$store.state.api_token,
853       },
854     };
855     this.$http
856       .post(this.$API_URL + "timeSlots/", params, config)
857       .then((response) => {
858         console.log(response);
859         this.loadEducatorScheduleOverridesWithTimeSlots();
860         this.$alertify.success("Créneau horaire ajouté avec succès");
861       })
862       .catch((error) => {
863         this.$alertify.error(error.response.data.error);
864       });
865   },
866   /**
867    * Method to create a weekly schedule from the api rest endpoint "POST api/v1/
868    * weeklySchedules".
869    *
870    * @param {string} dateValidFrom The date valid from of the weekly schedule
871    * @param {?string} dateValidTo The date valid to of the weekly schedule
872    */
873   createWeeklySchedule(dateValidFrom, dateValidTo) {
874     const params = new URLSearchParams();
875     params.append("date_valid_from", dateValidFrom);
876     if (dateValidTo !== null) {
877       params.append("date_valid_to", dateValidTo);
878     }
879     const config = {
880       headers: {
881         "Authorization" : this.$store.state.api_token,

```



```

880     },
881   };
882   this.$http
883     .post(this.$API_URL + "weeklySchedules/", params, config)
884     .then((response) => {
885       console.log(response);
886       this.loadEducatorWeeklySchedulesWithTimeSlots();
887       this.$alertify.success("Calendrier hebdomadaire ajouté avec succès");
888       this.$bvModal.hide("modal-add-weeklyschedule");
889     })
890     .catch((error) => {
891       this.$alertify.error(error.response.data.error);
892     });
893 },
894 /**
895  * Method to create a schedule override from the api rest endpoint "POST api/v1
896  * /scheduleOverrides".
897  *
898  * @param {string} dateScheduleOverride The date of schedule override
899  */
900 createScheduleOverride(dateScheduleOverride) {
901   const params = new URLSearchParams();
902   params.append("date_schedule_override", dateScheduleOverride);
903   const config = {
904     headers: {
905       "Authorization" : this.$store.state.api_token,
906     },
907   };
908   this.$http
909     .post(this.$API_URL + "scheduleOverrides/", params, config)
910     .then((response) => {
911       console.log(response);
912       this.loadEducatorScheduleOverridesWithTimeSlots();
913       this.$alertify.success("Exception d'horaire ajouté avec succès");
914       this.$bvModal.hide("modal-add-scheduleoverride");
915     })
916     .catch((error) => {
917       this.$alertify.error(error.response.data.error);
918     });
919 },
920 /**
921  * Method to create a absence from the api rest endpoint "POST api/v1/absences
922  * ".
923  *
924  * @param {string} dateAbsenceFrom The date absence from of absence
925  * @param {string} dateAbsenceTo The date absence to of absence
926  */
927 createAbsence(dateAbsenceFrom, dateAbsenceTo) {
928   const params = new URLSearchParams();
929   params.append("date_absence_from", dateAbsenceFrom);
930   params.append("date_absence_to", dateAbsenceTo);
931   const config = {
932     headers: {
933       "Authorization" : this.$store.state.api_token,
934     },
935   };
936   this.$http
937     .post(this.$API_URL + "absences/", params, config)
938     .then((response) => {

```

```

937         console.log(response);
938         this.loadEducatorAbsences();
939         this.$alertify.success("Vacances ajoutées avec succès");
940         this.$bvModal.hide("modal-add-absence");
941     })
942     .catch((error) => {
943         this.$alertify.error(error.response.data.error);
944     });
945 },
946 /**
947  * Method to delete a weekly schedule from his id from the api rest endpoint "
948  * DELETE api/v1/weeklySchedules/{weeklyScheduleId}"
949  *
950  * @param {string} id The weekly schedule id
951  */
952 deleteWeeklyScheduleById(id) {
953     const config = {
954         headers: {
955             "Authorization" : this.$store.state.api_token,
956         },
957     };
958     this.$http
959     .delete(this.$API_URL + "weeklySchedules/" + id, config)
960     .then((response) => {
961         this.loadEducatorWeeklySchedulesWithTimeSlots();
962         this.$alertify.success("Calendrier hebdomadaire supprimé");
963         console.log(response);
964     })
965     .catch((error) => {
966         this.$alertify.error(error.response.data.error);
967     });
968 },
969 /**
970  * Method to delete a schedule override from his id from the api rest endpoint
971  * "DELETE api/v1/scheduleOverrides/{scheduleOverrideId}"
972  *
973  * @param {string} id The schedule override id
974  */
975 deleteScheduleOverrideById(id) {
976     const config = {
977         headers: {
978             "Authorization" : this.$store.state.api_token,
979         },
980     };
981     this.$http
982     .delete(this.$API_URL + "scheduleOverrides/" + id, config)
983     .then((response) => {
984         this.loadEducatorScheduleOverridesWithTimeSlots();
985         this.$alertify.success("Exception d'horaire supprimé");
986         console.log(response);
987     })
988     .catch((error) => {
989         this.$alertify.error(error.response.data.error);
990     });
991 },
992 /**
993  * Method to delete a absence from his id from the api rest endpoint "DELETE
994  * api/v1/absences/{absenceId}"
995  *

```

```

993     * @param {string} id The absence id
994     */
995     deleteAbsenceById(id) {
996         const config = {
997             headers: {
998                 "Authorization" : this.$store.state.api_token,
999             },
1000         };
1001         this.$http
1002             .delete(this.$API_URL + "absences/" + id, config)
1003             .then((response) => {
1004                 this.loadEducatorAbsences();
1005                 this.$alertify.success("Vacances supprimées");
1006                 console.log(response);
1007             })
1008             .catch((error) => {
1009                 this.$alertify.error(error.response.data.error);
1010             });
1011     },
1012     /**
1013     * Method to delete a time slot from his id from the api rest endpoint "DELETE
1014     * api/v1/timeSlots/{timeSlotId}"
1015     *
1016     * @param {string} id The time slot id
1017     */
1018     deleteTimeSlotById(timeSlotId) {
1019         const config = {
1020             headers: {
1021                 "Authorization" : this.$store.state.api_token,
1022             },
1023         };
1024         this.$http
1025             .delete(this.$API_URL + "timeSlots/" + timeSlotId, config)
1026             .then((response) => {
1027                 this.loadEducatorWeeklySchedulesWithTimeSlots();
1028                 this.loadEducatorScheduleOverridesWithTimeSlots();
1029                 this.$alertify.success("Créneau horaire supprimé");
1030                 console.log(response);
1031             })
1032             .catch((error) => {
1033                 this.$alertify.error(error.response.data.error);
1034             });
1035     },
1036     sendWeeklyScheduleId(weeklyScheduleId) {
1037         this.selectedWeeklyScheduleId = weeklyScheduleId;
1038     },
1039     sendScheduleOverrideId(scheduleOverrideId) {
1040         this.selectedScheduleOverrideId = scheduleOverrideId;
1041     },
1042     sendTimeSlotId(timeSlotId) {
1043         this.selectedTimeSlotId = timeSlotId;
1044     },
1045     sendAbsenceId(absenceId) {
1046         this.selectedAbsenceId = absenceId;
1047     },
1048     formatDate(value) {
1049         if (value) {
1050             return moment(value).format("dddd DD MMMM YYYY");
1051         }
1052     }

```

```

1051     },
1052     formatTime(value) {
1053         if (value) {
1054             return moment("1970-01-01 " + value).format("HH[h]");
1055         }
1056     },
1057     clearWeeklyScheduleDateValidTo() {
1058         this.formWeeklySchedule.date_valid_to = null;
1059     },
1060 },
1061 mounted() {
1062     this.loadEducatorWeeklySchedulesWithTimeSlots();
1063     this.loadEducatorScheduleOverridesWithTimeSlots();
1064     this.loadEducatorAbsences();
1065 },
1066 };
1067 </script>
1068
1069 <style scoped>
1070 .title {
1071     margin-top: 20px;
1072     color: #3ea3d8;
1073 }
1074 .tableBtn {
1075     background-color: #3ea3d8;
1076 }
1077 </style>

```

Listing 16 – ./Sources/src/components/PrivacyPolicy.vue

```

1  <!--
2  PrivacyPolicy.vue
3
4  Component representing the privacy policies of the application.
5
6  Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10     <div class="content">
11         <b-container>
12             <b-row id="title" class="text-center">
13                 <b-col>
14                     <h1 class="font-weight-bold">Politique de confidentialité</h1>
15                 </b-col>
16             </b-row>
17             <b-row class="text-center">
18                 <b-col>
19                     <small>
20                         <p style="margin-bottom: 0px">
21                             Politique de confidentialité
22                         </p></small>
23                     >
24                     <small><p>Dernière révision : 15/04/2021</p></small>
25                 </b-col>
26             </b-row>
27             <b-row>
28                 <b-col id="privacyPolicy">
29                     <small>

```

```

30 <p>Introduction</p>
31 <p class="text-justify">
32     Nous nous soucions de la vie privée des utilisateurs de notre site
33     Internet et/ou de notre espace mobile (le « Site » et l'« Espace
34     mobile », respectivement) et nous nous engageons à protéger les
35     informations que les utilisateurs partagent avec nous lorsqu'ils
36     utilisent notre Site et/ou notre Espace mobile (collectivement,
37     les « Propriétés numériques »), et nous nous engageons pleinement
38     à protéger et à utiliser vos informations conformément à la loi
39     applicable. Cette Politique de confidentialité décrit nos
40     pratiques en matière de recueil, d'utilisation et de divulgation
41     de vos informations par le biais de nos propriétés numériques (les «
42     Services »), lorsque vous accédez aux Services depuis votre
43     appareil. Avant d'accéder ou d'utiliser l'un de nos Services,
44     veuillez lire attentivement cette Politique de confidentialité et
45     vous assurer que vous comprenez parfaitement nos pratiques
46     concernant vos informations. Si vous lisez et comprenez pleinement
47     cette Politique de confidentialité, et que vous restez opposé(e) à
48     nos pratiques, vous devez immédiatement et cesser toute
49     utilisation de nos Propriétés numériques et de nos Services. En
50     utilisant nos Services, vous acceptez les conditions de cette
51     Politique de confidentialité et le fait de continuer à utiliser
52     les Services constitue votre acceptation de cette Politique de
53     confidentialité et de tout amendement à celle-ci. Dans la présente
54     Politique de confidentialité, vous trouverez des explications sur
55     les questions suivantes :
56 </p>
57 <ul>
58     <li>Les informations que nous recueillons</li>
59     <li>Comment nous recueillons les informations</li>
60 </ul>
61 <p>
62     Les informations que nous recueillons<br />Vous trouverez
63     ci-dessous les différents types d'informations que nous pouvons
64     recueillir.
65 </p>
66 <ul>
67     <li>
68         Les informations non identifiées et non identifiables, qui
69         peuvent être fournies par vous lors du processus
70         d'enregistrement ou recueillies lorsque vous utilisez nos
71         Services (les « Informations non personnelles »). Les
72         informations non personnelles ne nous permettent pas
73         d'identifier la personne auprès de qui elles ont été
74         recueillies. Les Informations non personnelles que nous
75         recueillons consistent principalement en des informations
76         techniques et des informations d'utilisation agrégées.
77     </li>
78     <li>
79         Les informations permettant de vous identifier individuellement,
80         c'est-à-dire les informations qui vous identifient ou qui
81         peuvent, moyennant un effort raisonnable, vous identifier (les «
82         Informations personnelles »). Les Informations personnelles que
83         nous recueillons par le biais de nos Services sont notamment le
84         nom, l'adresse e-mail, l'adresse, le numéro de téléphone, ou
85         d'autres informations demandées à chaque fois que nécessaire. Si
86         nous combinons des Informations personnelles avec des
87         Informations non personnelles, nous traiterons les informations
88         combinées comme des Informations personnelles aussi longtemps

```

```

89         longtemps qu'elles resteront combinées.
90     </li>
91 </ul>
92 <p>
93     Comment nous recueillons les informations<br />Vous trouverez
94     ci-dessous les principales méthodes que nous utilisons pour
95     recueillir des informations :
96 </p>
97 <ul>
98     <li>
99         Nous recueillons des informations lorsque vous utilisez les
100        Services. Lorsque vous visitez nos Propriétés numériques ou que
101        vous utilisez nos Services, nous pouvons recueillir, collecter
102        et enregistrer ces utilisations, sessions et informations
103        connexes.
104     </li>
105     <li>
106        Nous recueillons les informations que vous nous fournissez
107        volontairement. Par exemple, nous recueillons les informations
108        que vous nous fournissez lorsque vous nous contactez directement
109        par n'importe quel canal de communication (par exemple lorsque
110        vous nous envoyez un e-mail contenant un commentaire ou un
111        retour d'information).
112     </li>
113 </ul>
114 </small>
115 </b-col>
116 </b-row>
117 </b-container>
118 </div>
119 </template>
120
121 <script>
122 export default {
123     name: "PrivacyPolicy",
124 };
125 </script>
126
127 <style scoped>
128 #title {
129     margin-top: 20px;
130     color: #3ea3d8;
131 }
132 </style>

```

Listing 17 – ./Sources/src/components/RightSectionContent.vue

```

1  <!--
2    RightSectionContent.vue
3
4    Reusable component representing an aesthetic content displayed with textual
      content on the right and an image on the left.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div class="content">
11      <b-container>

```

```

12     <b-row align-h="center">
13         <b-col
14             class="d-flex align-items-center justify-content-center"
15             cols="12"
16             sm="12"
17             lg="6"
18         >
19             <b-img
20                 v-if="imageSource != null"
21                 class="circleImage"
22                 :src="imageSource"
23                 fluid
24                 rounded="circle"
25                 alt="Circle image"
26             ></b-img>
27         </b-col>
28         <b-col cols="12" sm="12" lg="6">
29             <h1 class="font-weight-bold display-4">{{ textH1 }}</h1>
30             <h3 class="font-weight-bold">{{ textH3 }}</h3>
31             <h5>{{ textH5 }}</h5>
32             <h6>{{ textH6 }}</h6>
33             <p class="text-justify">{{ textParagraph }}</p>
34             <b-button id="button" pill :to="routeBouton" v-if="textButton != ''">
35                 {{ textButton }}</b-button>
36         >
37         </b-col>
38     </b-row>
39 </b-container>
40 </div>
41 </template>
42
43 <script>
44 export default {
45     props: {
46         textParagraph: {
47             type: String,
48             default: "",
49         },
50         textH1: {
51             type: String,
52             default: "",
53         },
54         textH3: {
55             type: String,
56             default: "",
57         },
58         textH5: {
59             type: String,
60             default: "",
61         },
62         textH6: {
63             type: String,
64             default: "",
65         },
66         routeBouton: {
67             type: String,
68             default: "",
69         },
70         textButton: {

```

```

71     type: String,
72     default: "",
73   },
74   imageSource: {
75     type: String,
76     default: "",
77   },
78 },
79 name: "RightSectionContent",
80 };
81 </script>
82
83 <style scoped>
84 .content {
85   background-color: #ffffff;
86   padding-top: 20px;
87   padding-bottom: 20px;
88   color: #3ea3d8;
89 }
90 #button {
91   background-color: #3ea3d8;
92 }
93 </style>

```

Listing 18 – ./Sources/src/components/SignaturePad.vue

```

1  <!--
2    PrivacyPolicy.vue
3
4    Reusable component representing the signature functionality.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  -->
8
9  <template>
10    <div class="content">
11      <b-container id="container">
12        <VueSignaturePad
13          id="pad"
14          :customStyle="customStyle"
15          ref="signaturePad"
16          :options="signaturePadOptions"
17        />
18        <b-row id="buttonRow" class="text-center">
19          <b-col>
20            <b-button
21              id="btnClearSignature"
22              block
23              type="button"
24              @click="clearSignaturePad"
25              variant="outline-danger"
26            >
27              Effacer
28            </b-button>
29          </b-col>
30          <b-col>
31            <b-button
32              id="btnValidateSignature"
33              :disabled="isBlocked"

```



```

34         block
35         type="button"
36         @click="getSignaturePad"
37         variant="outline-success"
38     >
39         Valider la signature
40     </b-button>
41 </b-col>
42 </b-row>
43 </b-container>
44 </div>
45 </template>
46
47 <script>
48 export default {
49     name: "SignaturePad",
50     data() {
51         return {
52             dataURL: "",
53             isBlocked: false,
54             customStyle: { border: "#c2c2c2 3px solid" },
55             signaturePadOptions: {
56                 onBegin: () => {
57                     this.$refs.signaturePad.resizeCanvas();
58                 },
59                 backgroundColor: "rgb(255, 255, 255)",
60             },
61         };
62     },
63     methods: {
64         getSignaturePad() {
65             this.isBlocked = true;
66             this.$refs.signaturePad.lockSignaturePad();
67             this.dataURL = this.$refs.signaturePad.saveSignature("image/jpeg").data;
68             this.$emit("getSignaturePad", this.dataURL, this.isBlocked);
69         },
70         clearSignaturePad() {
71             this.$refs.signaturePad.clearSignature();
72             this.isBlocked = false;
73             this.$refs.signaturePad.openSignaturePad();
74             this.$emit("getSignaturePad", this.dataURL, this.isBlocked);
75         },
76         resizeSignaturePadToSquare() {
77             this.$refs.signaturePad.resizeCanvas();
78             var padWidth = this.$jquery("#pad").width();
79             this.$jquery("#pad").css({ height: padWidth + "px" });
80         },
81     },
82     destroyed() {
83         window.removeEventListener("resize", this.resizeSignaturePadToSquare);
84     },
85     mounted() {
86         window.addEventListener("resize", this.resizeSignaturePadToSquare);
87         this.resizeSignaturePadToSquare();
88     },
89 };
90 </script>
91
92 <style scoped>

```

```

93 #container {
94     padding: 0px;
95 }
96 #buttonRow {
97     margin-top: 10px;
98     margin-bottom: 10px;
99 }
100 </style>

```

Listing 19 – ./Sources/src/constants.js

```

1  /*
2   constants.js
3
4   File containing the constants of the application.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  export const CUSTOMER_CODE_ROLE = "1";
10 export const ADMIN_CODE_ROLE = "2";

```

Listing 20 – ./Sources/src/main.js

```

1  /*
2   main.js
3
4   File allowing to initialize the main root component with all the configuration
   and the plugins necessary for the good functioning of the application.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  import "@babel/polyfill";
10 import "mutationobserver-shim";
11
12 import "./plugins/bootstrap-vue";
13 import "./plugins/vue-alertify";
14 import "./plugins/vue-signature-pad";
15
16 import "./assets/css/main.css";
17
18 import Vue from "vue";
19 import App from "./App.vue";
20 import router from "./router";
21 import store from "./store";
22
23 import Axios from "axios";
24 Vue.prototype.$http = Axios;
25 Vue.prototype.$API_URL = "https://api-rest-douceur-de-chien.boreljaquet.ch/v1/";
26
27 import jquery from "jquery";
28 Vue.prototype.$jquery = jquery;
29
30 new Vue({
31     router,
32     store,
33     render: (h) => h(App),
34 }).$mount("#app");

```

Listing 21 – ./Sources/src/plugins/bootstrap-vue.js

```

1  /*
2    bootstrap-vue.js
3
4    File allowing to include all the files necessary to the good functioning of
      bootstrapVue in Vue.js.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  import Vue from "vue";
10
11  import BootstrapVue from "bootstrap-vue";
12  import "bootstrap/dist/css/bootstrap.min.css";
13  import "bootstrap-vue/dist/bootstrap-vue.css";
14
15  Vue.use(BootstrapVue);

```

Listing 22 – ./Sources/src/plugins/vue-alertify.js

```

1  /*
2    vue-alertify.js
3
4    File allowing to include all the files necessary to the good functioning of vue-
      alertify in Vue.js.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  import Vue from "vue";
10
11  import VueAlertify from "vue-alertify";
12  Vue.use(VueAlertify);

```

Listing 23 – ./Sources/src/plugins/vue-signature-pad.js

```

1  /*
2    vue-signature-pad.js
3
4    File allowing to include all the files necessary to the good functioning of vue-
      signature-pad in Vue.js.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  import Vue from "vue";
10
11  import VueSignaturePad from "vue-signature-pad";
12  Vue.use(VueSignaturePad);

```

Listing 24 – ./Sources/src/router.js

```

1  /*
2    router.js
3
4    File to configure the application router with Vue Router.
5
6    Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */

```

```

8
9 import Vue from "vue";
10 import Router from "vue-router";
11 import store from "./store";
12 import Home from "./components/Home.vue";
13 import About from "./components/About.vue";
14 import Calendar from "./components/Calendar.vue";
15 import PrivacyPolicy from "./components/PrivacyPolicy.vue";
16 import Connection from "./components/Connection.vue";
17 import Inscription from "./components/Inscription.vue";
18 import CustomerInformation from "./components/CustomerInformation.vue";
19 import CustomerAppointment from "./components/CustomerAppointment.vue";
20 import Administration from "./components/Administration.vue";
21 import Planning from "./components/Planning.vue";
22 import EducatorCalendar from "./components/EducatorCalendar.vue";
23 import { CUSTOMER_CODE_ROLE, ADMIN_CODE_ROLE } from "./constants.js";
24
25 Vue.use(Router);
26
27 export default new Router({
28   routes: [
29     { path: "/", name: "home", component: Home },
30     { path: "/about", name: "about", component: About },
31     { path: "/calendar", name: "calendar", component: Calendar },
32     {
33       path: "/privacy_policy",
34       name: "privacyPolicy",
35       component: PrivacyPolicy,
36     },
37     {
38       path: "/connection",
39       name: "connection",
40       component: Connection,
41       beforeEnter(to, from, next) {
42         if (
43           store.state.api_token &&
44           store.state.code_role == CUSTOMER_CODE_ROLE
45         ) {
46           next("/customer_information");
47         } else {
48           next();
49         }
50         if (store.state.api_token && store.state.code_role == ADMIN_CODE_ROLE) {
51           next("/administration");
52         } else {
53           next();
54         }
55       },
56     },
57     {
58       path: "/inscription",
59       name: "inscription",
60       component: Inscription,
61       beforeEnter(to, from, next) {
62         if (
63           store.state.api_token &&
64           store.state.code_role == CUSTOMER_CODE_ROLE
65         ) {
66           next("/customer_information");

```

```

67         } else {
68             next();
69         }
70         if (store.state.api_token && store.state.code_role == ADMIN_CODE_ROLE) {
71             next("/administration");
72         } else {
73             next();
74         }
75     },
76 },
77 {
78     path: "/customer_information/:userId?",
79     name: "customerInformation",
80     component: CustomerInformation,
81     beforeEnter(to, from, next) {
82         if (store.state.api_token) {
83             next();
84         } else {
85             next("/");
86         }
87     },
88 },
89 {
90     path: "/customer_appointment/:userId?",
91     name: "customerAppointment",
92     component: CustomerAppointment,
93     beforeEnter(to, from, next) {
94         if (store.state.api_token) {
95             next();
96         } else {
97             next("/");
98         }
99     },
100 },
101 {
102     path: "/administration",
103     name: "administration",
104     component: Administration,
105     beforeEnter(to, from, next) {
106         if (store.state.api_token && store.state.code_role == ADMIN_CODE_ROLE) {
107             next();
108         } else {
109             next("/");
110         }
111     },
112 },
113 {
114     path: "/planning",
115     name: "planning",
116     component: Planning,
117     beforeEnter(to, from, next) {
118         if (store.state.api_token && store.state.code_role == ADMIN_CODE_ROLE) {
119             next();
120         } else {
121             next("/");
122         }
123     },
124 },
125 {

```

```

126     path: "/educator_calendar",
127     name: "educatorCalendar",
128     component: EducatorCalendar,
129     beforeEnter(to, from, next) {
130         if (store.state.api_token && store.state.code_role == ADMIN_CODE_ROLE) {
131             next();
132         } else {
133             next("/");
134         }
135     },
136 },
137 ],
138 });

```

Listing 25 – ./Sources/src/store.js

```

1  /*
2   store.js
3
4   File to configure the application store with Vuex.
5
6   Jonathan Borel-Jaquet - CFPT / T.IS-ES2 <jonathan.brljq@eduge.ch>
7  */
8
9  import Vue from "vue";
10 import Vuex from "vuex";
11 import createPersistedState from "vuex-persistedstate";
12 import router from "../router";
13 import { CUSTOMER_CODE_ROLE, ADMIN_CODE_ROLE } from "../constants.js";
14
15 Vue.use(Vuex);
16
17 export default new Vuex.Store({
18   plugins: [createPersistedState()],
19   state: {
20     api_token: null,
21     code_role: null,
22     user_id: null,
23   },
24   mutations: {
25     authUser(state, userData) {
26       state.api_token = userData.api_token;
27       state.code_role = userData.code_role;
28       state.user_id = userData.user_id;
29     },
30     clearAuth(state) {
31       state.api_token = null;
32       state.code_role = null;
33       state.user_id = null;
34     },
35   },
36   actions: {
37     inscription({ commit }, authData) {
38       const params = new URLSearchParams();
39       params.append("email", authData.email);
40       params.append("firstname", authData.firstname);
41       params.append("lastname", authData.lastname);
42       params.append("phonenummer", authData.phonenumber);
43       params.append("address", authData.address);

```

```

44     params.append("password", authData.password);
45     params.append(
46         "reCAPTCHAuserResponseToken",
47         authData.reCAPTCHAuserResponseToken
48     );
49     Vue.prototype.$http
50     .post(Vue.prototype.$API_URL + "users/", params)
51     .then((res) => {
52         console.log(res);
53         localStorage.setItem("api_token", res.data.api_token);
54         localStorage.setItem("code_role", res.data.code_role);
55         localStorage.setItem("user_id", res.data.user_id);
56         commit("authUser", {
57             api_token: res.data.api_token,
58             code_role: res.data.code_role,
59             user_id: res.data.user_id,
60         });
61         Vue.prototype.$alertify.success("Vous êtes inscrit");
62         router.push("/customer_information");
63     })
64     .catch((error) => {
65         Vue.prototype.$alertify.error(error.response.data.error);
66     });
67 },
68 login({ commit }, authData) {
69     const params = new URLSearchParams();
70     params.append("email", authData.email);
71     params.append("password", authData.password);
72     Vue.prototype.$http
73     .post(Vue.prototype.$API_URL + "connection/", params)
74     .then((res) => {
75         localStorage.setItem("api_token", res.data.api_token);
76         localStorage.setItem("code_role", res.data.code_role);
77         localStorage.setItem("user_id", res.data.user_id);
78         commit("authUser", {
79             api_token: res.data.api_token,
80             code_role: res.data.code_role,
81             user_id: res.data.user_id,
82         });
83         Vue.prototype.$alertify.success("Vous êtes connecté");
84         if (this.state.code_role == ADMIN_CODE_ROLE) {
85             router.push("/administration");
86         }
87         if (this.state.code_role == CUSTOMER_CODE_ROLE) {
88             router.push("/customer_information");
89         }
90     })
91     .catch((error) => {
92         Vue.prototype.$alertify.error(error.response.data.error);
93     });
94 },
95 logout({ commit }) {
96     commit("clearAuth");
97     localStorage.removeItem("api_token");
98     localStorage.removeItem("code_role");
99     localStorage.removeItem("user_id");
100     Vue.prototype.$alertify.success("Vous êtes déconnecté");
101     router.push("/connection");
102 },

```

```

103     autoLogin({ commit }) {
104         const api_token = localStorage.getItem("api_token");
105         const code_role = localStorage.getItem("code_role");
106         const user_id = localStorage.getItem("user_id");
107         if (!api_token || !code_role || !user_id) {
108             return;
109         }
110         commit("authUser", {
111             api_token: api_token,
112             code_role: code_role,
113             user_id: user_id,
114         });
115     },
116 },
117 getters: {
118     ifCustomerAuthenticated(state) {
119         return state.api_token !== null && state.code_role == CUSTOMER_CODE_ROLE;
120     },
121     ifAdministratorAuthenticated(state) {
122         return state.api_token !== null && state.code_role == ADMIN_CODE_ROLE;
123     },
124     ifAuthenticated(state) {
125         return state.api_token !== null;
126     },
127 },
128 });

```