

```

<template>
  <div class="details">
    <div class="details__header">
      <div class="information">
        <Badge
          v-if="hasEmptyFields"
          size="small"
        />
        <p class="t-md semibold">
          Additional Fields
        </p>
      </div>
      <div class="icon-container c-tertiary">
        <i class="mdi mdi-list-box-outline" />
        <p>
          {{ localApplicantContractFields.length }}
          {{ localApplicantContractFields.length > 1 ? 'fields' : 'field' }}
        </p>
      </div>
      <DisclosureButton
        alternate
        :toggled="isOpen"
        @toggle-state="isOpen = !isOpen"
      />
    </div>
    <ul
      v-show="isOpen"
      class="details__body"
    >
      <li
        v-for="(field, index) in localApplicantContractFields"
        :key="`other-field-${index}`"
        class="field-item"
      >
        <div class="label">
          <p class="required-field">
            {{ getAddressLabel(field) }}
          </p>
        </div>
        <div class="value">
          <template v-if="field.name.includes('address')">
            <div>
              <DropDownButton
                :label="countryLabel ? countryLabel : 'Select Country'"
            </div>

```

```

class="block"
:class="{invalid: isEmpty(addressParts.country)
  && !fieldErrors['applicant-address']}"
:disabled="isUpdatingContract || !hasApplicantFillableFields"
>
  <DropdownMenu
    class="block"
    search-bar
    :override-close-dropdown="!filteredCountries.length"
    placeholder="Search country"
    @search="countrySearchKey = $event"
  >
    <ul>
      <template v-if="filteredCountries.length">
        <li
          v-for="country in filteredCountries"
          :key="'${country.id}-${country.pk}'"
          @click="updateAddressPart('country', country)"
        >
          {{ country.name }}
        </li>
      </template>
      <li v-else class="not-selectable">
        <p class="c-tertiary">
          No results found
        </p>
      </li>
    </ul>
  </DropdownMenu>
</DropdownButton>
</div>
<div>
  <DropdownButton
    :label="provinceLabel ? provinceLabel : 'Select Province'"
    class="block"
    :class="{invalid: isEmpty(addressParts.province)
      && !fieldErrors['applicant-address']}"
    :disabled="!countryLabel || !hasProvinces
      || isUpdatingContract || !hasApplicantFillableFields"
  >
    <DropdownMenu
      class="block"
      search-bar
      :override-close-dropdown="!filteredProvinces.length"

```

```

        placeholder="Search province"
        @search="provinceSearchKey = $event"
    >
    <ul>
        <template v-if="filteredProvinces.length">
            <li
                v-for="province in filteredProvinces"
                :key="'${province.id}-${province.pk}'"
                @click="updateAddressPart('province', province)"
            >
                {{ province.name }}
            </li>
        </template>
        <li v-else class="not-selectable">
            <p class="c-tertiary">
                No results found
            </p>
        </li>
    </ul>
    </DropdownMenu>
</DropdownButton>
</div>
<div>
    <input
        id="street-address"
        v-model="streetLabel"
        class="block"
        type="text"
        placeholder="Enter Street Address"
        :disabled="isUpdatingContract || !hasApplicantFillableFields"
        @input="updateAddressPart('street', $event.target.value)"
    />
</div>
</template>
<template v-else-if="field.name.includes('civil-status')">
    <DropdownButton
        :label="field.value || 'Select civil status'"
        class="block"
        :class="{ 'invalid': isFieldEmpty(field.value)
            && !!fieldErrors[field.name]}"
        :disabled="isUpdatingContract || !hasApplicantFillableFields"
    >
        <DropdownMenu class="block">
            <ul>

```

```

        <li
            v-for="status in civilStatuses"
            :key="status.id + status.value"
            :value="status.value"
            @click="updateFieldValue(field, status.value)"
        >
            {{ status.value }}
        </li>
    </ul>
</DropDownMenu>
</DropDownButton>
<p
    v-if="!field.value"
    class="t-sm c-error"
>
    {{ fieldErrors[field.name] }}
</p>
</template>
<template v-else>
    <input
        v-model="field.value"
        class="block"
        type="text"
        :placeholder="setPlaceholder(field.placeholder)"
        :class="{ 'invalid': isFieldEmpty(field.value)
            && !!fieldErrors[field.name] }"
        :disabled="isUpdatingContract || !hasApplicantFillableFields"
        @input="updateFieldValue(field, $event.target.value)"
    />
    <p
        v-if="!field.value"
        class="t-sm c-error"
    >
        {{ fieldErrors[field.name] }}
    </p>
</template>
</div>
</li>
<Button
    v-if="localApplicantContractFields.length && hasApplicantFillableFields"
    :class="" primary
    ${windowWidth <= MOBILE ? 'block small' : ''}
    `
    :disabled="isUpdatingContract || isUpdateButtonDisabled"

```

```

        @click="updateApplicantContractFields"
      >
        Update
      </Button>
    </ul>
  </div>
</template>

<script>
import { mapGetters, mapActions } from 'vuex';
import Button from 'source/components/_generics/Button.vue';
import DisclosureButton from 'source/components/_generics/DisclosureButton.vue';
import DropdownButton from 'source/components/_generics/DropdownButton.vue';
import DropdownMenu from 'source/components/_generics/DropdownMenu.vue';
import CompensationConversionMixin from 'source/mixins/CompensationConversionMixin.js';
import CurrencyMixin from 'source/mixins/CurrencyMixin.js';
import EditorMirrorFieldMixin from 'source/mixins/EditorMirrorFieldMixin.js';
import Badge from 'source/components/_generics/Badge.vue';

export default {
  name: 'ContractFields',

  components: {
    DisclosureButton,
    Button,
    DropdownButton,
    DropdownMenu,
    Badge
  },

  mixins: [
    EditorMirrorFieldMixin,
    CurrencyMixin,
    CompensationConversionMixin
  ],

  props: {
    selectedContract: {
      type: Object,
      required: true
    }
  },

```