

```

        Non-exempt
    </p>
    <p class="t-sm c-tertiary">
        Entitled to Premium benefits
        (Premium rates for OT, ND, and Holiday)
    </p>
</li>
</ul>
</DropDownMenu>
</DropDownButton>
<p
    v-else
    class="block static-display"
>
    {{ employmentRule.type }}
</p>
</div>
<div class="field-container">
    <p
        class="c-secondary"
        :class="!isAddContract ? 'required-field' : ''"
    >
        Work Schedule
    </p>
    <div
        v-if="!jobOffer.schedule.name"
        class="work-sched job-desc"
        @click="clickSelectSchedule"
    >
        <i class="mdi mdi-plus-circle c-accent"></i>
        <p class="bold c-accent">
            Select Schedule
        </p>
    </div>
    <div
        v-else
        class="work-sched has-value"
        :class="!isAddContract ? 'job-desc' : ''"
    >
        <p
            :class="isAddContract ? 'static-display' : 'bold c-secondary'"
        >
            {{ jobOffer.schedule.name }}
        </p>
    </div>
    </div>
    </div>

```

```

        :class="!isAddContract ? 'required-field' : ''"
    >
        {{ compensationLabel }}
    </p>
    <input
        v-if="!isAddContract"
        v-model="jobOffer.compensationValue"
        v-limit-character-size.number="11"
        :disabled="viewMode"
        :placeholder="employmentRule.payableType"
        class="number"
        type="number"
    />
    <p
        v-else
        class="block static-display"
    >
        {{ jobOffer.compensationValue }}
    </p>
</div>
</div>
</div>
<div class="group">
    <div class="group-header">
        <p class="header-title bold c-secondary">
            Overtime
        </p>
    </div>
    <div class="group-body over-time">
        <div class="field-container">
            <div class="field-two-grid">
                <p class="c-secondary text-container">
                    Allow Overtime
                </p>
                <DropDownButton
                    v-if="!isAddContract"
                    :label="employmentRule.allowOvertime"
                    :disabled="viewMode"
                >
                    <DropDownMenu>
                        <ul>
                            <li @click="employmentRule.allowOvertime = 'No'">
                                No

```

```

    />
  </div>
</template>

<script>
  import { mapActions, mapState } from 'vuex';
  import { shallowRef } from 'vue';

  // components
  import Editor from 'source/components/editor/Editor.vue';
  import BaseCard from 'source/components/_generics/BaseCard.vue';
  import BaseAlert from 'source/components/_generics/BaseAlert.vue';
  import Button from 'source/components/_generics/Button.vue';
  import Tooltip from 'source/components/_generics/Tooltip.vue';
  import BaseCurrency from 'source/components/_generics/BaseCurrency.vue';
  import DropdownButton from 'source/components/_generics/DropdownButton.vue';
  import DropdownMenu from 'source/components/_generics/DropdownMenu.vue';
  import SelectContractTemplateModal from
'source/components/employers/applicants/SelectContractTemplateModal.vue';
  import SelectJobDescriptionModal from
'source/components/employers/applicants/SelectJobDescriptionModal.vue';
  import JobDescriptionViewerModal from
'source/components/employers/applicants/JobDescriptionViewerModal.vue';
  import EditorToolbar from 'source/components/editor/EditorToolbar.vue';
  import ContractOptions from 'source/pages/employers/contracts/ContractOptions.vue';
  import SelectEmploymentPolicyModal from
'source/components/employers/applicants/SelectEmploymentPolicyModal.vue';
  import DatePicker from 'source/components/_generics/DatePicker.vue';
  import LoadingSpinner from 'source/components/_generics/LoadingSpinner.vue';
  import SelectWorkScheduleModal from
'source/components/employers/job-postings/popups/SelectWorkScheduleModal.vue';

  // mixins
  import TimezoneMixin from 'source/mixins/TimezoneMixin.js';
  import CurrencyMixin from 'source/mixins/CurrencyMixin.js';
  import EditorMirrorFieldMixin from 'source/mixins/EditorMirrorFieldMixin.js';
  import FieldValidatorMixin from 'source/mixins/FieldValidatorMixin.js';

  // directives
  import { StripNonNumericCharacters } from 'source/directives/StripNonNumericCharacters.js';
  import { LimitCharacterSize } from 'source/directives/LimitCharacterSize.js';
  import { MousedownOutside } from 'source/directives/MousedownOutside.js';

  export default {

```

```

        this.$refs.headerPicker.click();
    },

    handleImageUpload(event) {
        this.$refs.toolbar.handleImageUpload(event);
    },

    toggleDescDropdown() {
        this.isDescDropdownOpen = !this.isDescDropdownOpen;
    },

    toggleScheduleDropdown() {
        this.isScheduleDropdownOpen = !this.isScheduleDropdownOpen;
    },

    closeDescDropdown() {
        this.isDescDropdownOpen = false;
    },

    closeScheduleDropdown() {
        this.isScheduleDropdownOpen = false;
    },

    removeSchedule() {
        this.jobOffer.schedule.pk = -1;
        this.jobOffer.schedule.name = "";
        this.isScheduleDropdownOpen = !this.isScheduleDropdownOpen;
    },

    saveContractHandler() {
        if (this.hasEmptyField()) {
            return;
        }

        try {
            this.validateQuillDocumentNotEmpty(this.jobOffer.contractContent);
            this.validateNotEmpty('Job Position', this.jobOffer.jobTitle);
            this.validateNotNegative(this.jobOffer.schedule.pk);
            const dataToValidate = {
                'Start Date': this.jobOffer.startDate,
                'Compensation Value': this.jobOffer.compensationValue,
            };
            const hasProjectName = this.otherFields?.filter((field) => {
                return field.insert.input.name.includes('project-name');
            });
        }
    }

```