

```
<template>
  <Page>
    <template v-if="isLoading">
      <div class="information">
        <BaseLoading
          height="30px"
          width="200px"
          border-radius="4px"
        />
        <BaseLoading
          height="24px"
          width="100px"
          border-radius="4px"
        />
      </div>
    </template>
    <template v-else>
      <div class="header-wrapper">
        <BackButton :route="previousRoute" />
        <div class="header-description">
          <p class="p-h1 c-primary">
            Contract Workflow
          </p>
        </div>
      </div>
    </template>
    <AlertBanner
      v-if="!isLoading"
      icon="information-outline"
      :label="contractWorkflowInfo"
      color="blue"
    />
    <div class="body">
      <template v-if="isLoading">
        <BaseLoading
          height="600px"
          width="1200px"
          border-radius="4px"
        />
        <div class="add-template">
```

```

        <BaseLoading
            height="40px"
            width="200px"
            border-radius="4px"
        />
    </div>
</template>
<template v-else>
    <template
        v-for="(contract, index) in nextContracts"
    >
        <EmployeeFirstContract
            v-if="contract.pk && (contract.date_signed ||
contract.is_active)"
            :key="contract.pk || contract.contractPk || index"
            :contract="contract"
            :contract-policy="contract.contractPolicy"
        />
        <ContractConfig
            v-else
            :key="contract.pk || contract.contractPk"
            v-model="nextContracts[index]"
            :is-for-release="isForRelease"
            :is-editable="true"
            :applicant="applicant"
            :start-date="nextContracts[0].start_date"
            :is-saving-contract="isSubmitting"
            :contract-saved="contractSaved"
            :contract-edited="contractEdited"
            @save-contract="openSignatureModal(index)"
            @delete-contract="deleteContractHandler(index)"
        />
    </template>
    <div class="add-template">
        <Button
            v-if="showAddContractButton"
            :disabled="disableAddContractButton"
            class="large border primary bold"
            icon="plus-circle"
            @click="addContractHandler()"
        />
    </div>
</template>

```

```

        >
        Create another contract
      </Button>
    </div>
  </template>
</div>
<!-- <CreateScheduleModal
  v-if="createSchedule"
  :id="-1"
  :company-id="companyId"
  @close-modal="createSchedule = false"
  @save-schedule="saveSchedule"
/> -->
<SignatureModal
  v-if="isSignatureOpen"
  @close="closeSignatureModal"
  @confirm-signature="createContract"
/>
</Page>
</template>

<script>
  import Page from 'source/components/_generics/layouts/Page.vue';
  import Button from 'source/components/_generics/Button.vue';
  import BackButton from 'source/components/_generics/BackButton.vue';
  import ContractConfig from
'source/pages/employers/contracts/ContractConfig.vue';
  import EmployeeFirstContract from
'source/components/employers/employees/EmployeeFirstContract.vue';
  // import CreateScheduleModal from
'source/components/employers/work-schedule/CreateScheduleModal.vue';
  import HideFooterMixin from 'source/mixins/HideFooterMixin.js';
  import CurrencyMixin from 'source/mixins/CurrencyMixin.js';
  import FieldValidatorMixin from 'source/mixins/FieldValidatorMixin.js';
  import { mapActions, mapState } from 'vuex';
  import SignatureModal from
'source/components/_generics/SignatureModal.vue';
  import UploadFileMixin from 'source/mixins/UploadFileMixin.js';
  import BaseLoading from 'source/components/_generics/BaseLoading.vue';
  import AlertBanner from 'source/components/_generics/AlertBanner.vue';

```

```
const RANK_AND_FILE = 1;
const SUPERVISOR = 2;
const MANAGER = 3;

const EXEMPT = 1;
const NONEXEMPT = 2;

const HOURLY = 1;
const MONTHLY = 2;

export default {
  name: 'ContractWorkflowPage',

  components: {
    Page,
    Button,
    BackButton,
    ContractConfig,
    EmployeeFirstContract,
    // CreateScheduleModal,
    SignatureModal,
    BaseLoading,
    AlertBanner,
  },

  mixins: [
    HideFooterMixin,
    CurrencyMixin,
    FieldValidatorMixin,
    UploadFileMixin,
  ],

  provide() {
    return {
      getAddContract: () => this.isAddContract,
      getContractId: () => this.prevContract.pk
    };
  },
}
```

```
data() {
  return {
    isCampaignPolicyOpen: false,
    applicant: null,
    nextContracts: [],
    isSubmitting: false,
    isLoading: false,
    // createSchedule: false,
    // jobScheduleGroups: [],
    contractSaved: false,
    contractEdited: false,
    isAddContract: false,
    disableAddContractButton: false,
    prevContract: [],
    isSignatureOpen: false,
    selectedIndex: null,
    signatureDetails: null,
  };
},

computed: {

  previousRoute() {
    return {
      name: 'staff-employee-details'
    };
  },

  isForRelease() {
    return !!this.$route.query.contract_id;
  },

  companyId() {
    return this.$route.params.company_id;
  },

  isOrgExecutive() {
    return this.user?.user_profile?.is_organizational_exec;
```