

```

import { useState } from "react";
import axios from "axios";
import Header from "../components/Layout/Header";
import useAuth from "../context/useAuth";
import useCompany from "../context/useCompany";

export default function AppointmentLetterPage() {
  const { user } = useAuth();
  const { selectedCompany } = useCompany();

  const [loading, setLoading] = useState(false);

  const [form, setForm] = useState({
    // Document Details
    documentName: "",
    irn: "",

    // Employee Details
    employeeName: "",
    address: "",
    contactNumber: "",
    emailId: "",
    dateOfBirth: "",

    // Job Details
    jobTitle: "",
    joiningDate: "",
    workLocation: "",

    // Compensation Details
    basicMonthly: "",
    basicAnnual: "",
    hraMonthly: "",
    hraAnnual: "",
    specialAllowanceMonthly: "",
    specialAllowanceAnnual: "",
    medicalExpensesMonthly: "",
    medicalExpensesAnnual: "",
    educationalAllowanceMonthly: "",
    educationalAllowanceAnnual: "",
    conveyanceMonthly: "",
    conveyanceAnnual: "",
    performanceLinkMonthly: "",
    performanceLinkAnnual: "",
    providentFundMonthly: "",
    providentFundAnnual: "",
  });

```

```

    gratuityMonthly: "",
    gratuityAnnual: "",
    totalMonthly: "",
    totalAnnual: "",
    totalFixedCTCMonthly: "",
    totalFixedCTCAnnual: "",
    monthlyInHand: "",
    ctcAmount: "",
    ctcEffectiveDate: "",

    // Signatory Details
    authorizedSignatoryName: "",
    authorizedSignatoryDesignation: "",

    // Annexure Details
    annexureEmployeeName: "",
    annexureDesignation: "",
    annexureDOJ: "",
    annexureCTC: "",
    annexureCTCEffectiveDate: "",

    // Invitees
    invitee1Name: "",
    invitee1Email: "",
    invitee1Phone: "",

    invitee2Name: "",
    invitee2Email: "",
    invitee2Phone: "",

    invitee3Name: "",
    invitee3Email: "",
    invitee3Phone: "",

    invitee4Name: "",
    invitee4Email: "",
    invitee4Phone: "",
  });

  function handleChange(e) {
    setForm((prev) => ({
      ...prev,
      [e.target.name]: e.target.value,
    }));
  }

  async function handleSubmit(e) {
    e.preventDefault();
  }

```

```

setLoading(true);

try {
  const payload = {
    file: {
      name: form.documentName,

      fields: [
        {
          id: "1774612935369",
          name: "Full name of the employee",
          type: "text",
          value: form.employeeName,
          required: false,
        },
        {
          id: "1774612950757",
          name: "Residential Address of the employee",
          type: "text",
          value: form.address,
          required: false,
        },
        {
          id: "1774612969002",
          name: "Contact number of employee",
          type: "text",
          value: form.contactNumber,
          required: false,
        },
        {
          id: "1774612979379",
          name: "Email id of employee",
          type: "text",
          value: form.emailId,
          required: false,
        },
        {
          id: "1774613087824",
          name: "Job title/role offered",
          type: "text",
          value: form.jobTitle,
          required: false,
        },
        {
          id: "1774613074391",
          name: "Full name of employee",
          type: "text",

```

```

        value: form.employeeName,
        required: false,
    },
    {
        id: "1774614525025",
        name: "Job title/role offered of the employee",
        type: "text",
        value: form.jobTitle,
        required: false,
    },
    {
        id: "1774614525026",
        name: "Employee's joining date",
        type: "text",
        value: form.joiningDate,
        required: false,
    },
    {
        id: "1774620180774",
        name: "Work location/office",
        type: "text",
        value: form.workLocation,
        required: false,
    },
    {
        id: "1774619964580",
        name: "Employee's date of birth",
        type: "text",
        value: form.dateOfBirth,
        required: false,
    },
    {
        id: "1774620180777",
        name: "Authorized Signatory Name",
        type: "text",
        value: form.authorizadSignatoryName,
        required: false,
    },
    {
        id: "1774620180776",
        name: "Employee Name under signature part",
        type: "text",
        value: form.employeeName,
        required: false,
    },
    {
        id: "1774620180778",
        name: "Authorized Signatory Designation",

```

```

        type: "text",
        value: form.authorizadSignatoryDesignation,
        required: false,
    },
    {
        id: "1774620180780",
        name: "Employee Name (Annexure-I)",
        type: "text",
        value: form.annexureEmployeeName,
        required: false,
    },
    {
        id: "1774620180781",
        name: "Employee Designation (Annexure-I)",
        type: "text",
        value: form.annexureDesignation,
        required: false,
    },
    {
        id: "1774620180782",
        name: "Employee DOJ (Annexure-I)",
        type: "text",
        value: form.annexureDOJ,
        required: false,
    },
    {
        id: "1774620180783",
        name: "Employee CTC in Rs. (Annexure-I)",
        type: "text",
        value: form.annexureCTC,
        required: false,
    },
    {
        id: "1774620180784",
        name: "Employee CTC w.e.f. (Annexure-I)",
        type: "text",
        value: form.annexureCTCEffectiveDate,
        required: false,
    },
    {
        id: "1774620180785",
        name: "Basic (Monthly CTC)",
        type: "text",
        value: form.basicMonthly,
        required: false,
    },
    {
        id: "1774861862279",

```

```

        name: "Basic (Annual CTC)",
        type: "text",
        value: form.basicAnnual,
        required: false,
    },
    {
        id: "1774620180787",
        name: "HRA (Monthly CTC)",
        type: "text",
        value: form.hraMonthly,
        required: false,
    },
    {
        id: "1774861862281",
        name: "HRA (Annual CTC)",
        type: "text",
        value: form.hraAnnual,
        required: false,
    },
    {
        id: "1774620180789",
        name: "Special Allowance (Monthly CTC)",
        type: "text",
        value: form.specialAllowanceMonthly,
        required: false,
    },
    {
        id: "1774861862283",
        name: "Special Allowance (Annual CTC)",
        type: "text",
        value: form.specialAllowanceAnnual,
        required: false,
    },
    {
        id: "1774620180791",
        name: "Medical Expenses (Monthly CTC)",
        type: "text",
        value: form.medicalExpensesMonthly,
        required: false,
    },
    {
        id: "1774861862285",
        name: "Medical Expenses (Annual CTC)",
        type: "text",
        value: form.medicalExpensesAnnual,
        required: false,
    },
    {

```

```

        id: "1774620180793",
        name: "Educational Allowance (Monthly CTC)",
        type: "text",
        value: form.educationalAllowanceMonthly,
        required: false,
    },
    {
        id: "1774861862287",
        name: "Educational Allowance (Annual CTC)",
        type: "text",
        value: form.educationalAllowanceAnnual,
        required: false,
    },
    {
        id: "1774620180795",
        name: "Conveyance (Monthly CTC)",
        type: "text",
        value: form.conveyanceMonthly,
        required: false,
    },
    {
        id: "1774861862289",
        name: "Conveyance (Annual CTC)",
        type: "text",
        value: form.conveyanceAnnual,
        required: false,
    },
    {
        id: "1774620180797",
        name: "Performance Link Incentive (Monthly CTC)",
        type: "text",
        value: form.performanceLinkMonthly,
        required: false,
    },
    {
        id: "1774861862291",
        name: "Performance Link Incentive (Annual CTC)",
        type: "text",
        value: form.performanceLinkAnnual,
        required: false,
    },
    {
        id: "1774620180799",
        name: "Provident Fund (Monthly CTC)",
        type: "text",
        value: form.providentFundMonthly,
        required: false,
    },

```

```

{
  id: "1774861862295",
  name: "Provident Fund (Annual CTC)",
  type: "text",
  value: form.providentFundAnnual,
  required: false,
},
{
  id: "1774620180801",
  name: "Gratuity (Monthly CTC)",
  type: "text",
  value: form.gratuityMonthly,
  required: false,
},
{
  id: "1774861862297",
  name: "Gratuity (Annual CTC)",
  type: "text",
  value: form.gratuityAnnual,
  required: false,
},
{
  id: "1774620180803",
  name: "Total (Monthly CTC)",
  type: "text",
  value: form.totalMonthly,
  required: false,
},
{
  id: "1774861862299",
  name: "Total (Annual CTC)",
  type: "text",
  value: form.totalAnnual,
  required: false,
},
{
  id: "1774620180805",
  name: "Total Fixed CTC (A+B) (Monthly CTC)",
  type: "text",
  value: form.totalFixedCTCMonthly,
  required: false,
},
{
  id: "1774861862301",
  name: "Total Fixed CTC (A+B) (Annual CTC)",
  type: "text",
  value: form.totalFixedCTCAnnual,
  required: false,
}

```



```

    },
    {
      id: "1774620180807",
      name: "Monthly in Hand",
      type: "text",
      value: form.monthlyInHand,
      required: false,
    },
    {
      id: "1774614525027",
      name: "Employee Name ",
      type: "text",
      value: form.employeeName,
      required: false,
    },
  ],
},
invitees: [
  {
    name: form.invitee1Name,
    email: form.invitee1Email,
    phone: form.invitee1Phone,
  },
  {
    name: form.invitee2Name,
    email: form.invitee2Email,
    phone: form.invitee2Phone,
  },
  {
    name: form.invitee3Name,
    email: form.invitee3Email,
    phone: form.invitee3Phone,
  },
  {
    name: form.invitee4Name,
    email: form.invitee4Email,
    phone: form.invitee4Phone,
  },
].filter((i) => i.name),

irn: form.irn,
};

const response = await axios.post(
  "http://localhost:5000/api/sign/request?type=appointment",
  payload,
  {

```

```

        headers: {
            Authorization: `Bearer ${user.token}`,
            "X-Company": selectedCompany,
        },
    }
);

alert("Appointment Letter Created Successfully");

console.log(response.data);
} catch (err) {
    console.error(err);

    alert(
        err.response?.data?.message ||
        "Unable to create signing request."
    );
}

setLoading(false);
}

return (
    <>
        <Header
            title="Send Appointment Letter"
            subtitle="Create and send an appointment letter for electronic
signing"
            backDisable={false}
        />

        <div className="container-fluid px-3 px-lg-4 py-3">

            <form onSubmit={handleSubmit}>

                <div className="card border-0 shadow-sm">

                    { /* PAGE HEADER */ }
                    <div className="card-header bg-white border-bottom px-4
py-3">

                        <div className="d-flex justify-content-between
align-items-center">

                            <div>
                                <h5 className="mb-1 fw-semibold">
                                    Appointment Letter
                                </h5>

```

```

        <small className="text-muted">
            Enter employee, compensation and
signing details
        </small>
    </div>

    <span
className="badge bg-light text-dark border px-3 py-2">
        {selectedCompany}
    </span>

</div>

</div>

<div className="card-body p-3 p-lg-4">

    <div className="row g-4">

        {/*
===== */}
        {/* LEFT COLUMN */}
        {/*
===== */}

        <div className="col-lg-5">

            {/* DOCUMENT */}
            <SectionTitle
                number="01"
                title="Document Details"
            />

            <div className="row g-2 mb-4">

                <div className="col-md-7">
                    <FormField
                        label="Document Name"
                        name="documentName"
                        value={form.documentName}
                        onChange={handleChange}
                        placeholder="Document name"
                        required
                    />
                </div>

                <div className="col-md-5">

```

```

        <FormField
            label="IRN"
            name="irn"
            value={form.irn}
            onChange={handleChange}
            placeholder="Optional"
        />
    </div>

</div>

{ /* EMPLOYEE */ }
<SectionTitle
    number="02"
    title="Employee Details"
/>

<div className="row g-2 mb-4">

    <div className="col-md-6">
        <FormField
            label="Full Name"
            name="employeeName"
            value={form.employeeName}
            onChange={handleChange}
            placeholder="Employee name"
            required
        />
    </div>

    <div className="col-md-6">
        <FormField
            label="Email ID"
            name="emailId"
            type="email"
            value={form.emailId}
            onChange={handleChange}
            placeholder="Email address"
        />
    </div>

    <div className="col-md-6">
        <FormField
            label="Contact Number"
            name="contactNumber"
            value={form.contactNumber}
            onChange={handleChange}

```

residential address"

```
        placeholder="Phone number"
    />
</div>

<div className="col-md-6">
    <FormField
        label="Date of Birth"
        name="dateOfBirth"
        type="date"
        value={form.dateOfBirth}
        onChange={handleChange}
    />
</div>

<div className="col-12">
    <FormField
        label="Residential Address"
        name="address"
        value={form.address}
        onChange={handleChange}
        placeholder="Employee

        textarea
        rows={2}
    />
</div>

</div>

{/* JOB */}
<SectionTitle
    number="03"
    title="Job Details"
/>

<div className="row g-2 mb-4">

    <div className="col-md-6">
        <FormField
            label="Job Title / Role"
            name="jobTitle"
            value={form.jobTitle}
            onChange={handleChange}
            placeholder="Job title"
        />
    </div>
</div>
```

```

<div className="col-md-6">
  <FormField
    label="Joining Date"
    name="joiningDate"
    type="date"
    value={form.joiningDate}
    onChange={handleChange}
  />
</div>

<div className="col-12">
  <FormField
    label="Work Location / Office"
    name="workLocation"
    value={form.workLocation}
    onChange={handleChange}
    placeholder="Office / location"
  />
</div>

</div>

{ /* SIGNATORY */ }
<SectionTitle
  number="05"
  title="Signatory Details"
/>

<div className="row g-2 mb-4">

  <div className="col-md-6">
    <FormField
      label="Authorized Signatory"
      name="authorizadSignatoryName"
      value={
        form.authorizadSignatoryName
      }
      onChange={handleChange}
      placeholder="Full name"
    />
  </div>

  <div className="col-md-6">
    <FormField
      label="Designation"
      name="authorizadSignatoryDesignation"
    />
  </div>
</div>

```

```

        value={
form.authorizadSignatoryDesignation
        }
        onChange={handleChange}
        placeholder="Designation"
    />
</div>

</div>

{ /* ANNEXURE */ }
<SectionTitle
    number="06"
    title="Annexure-I Details"
/>

<div className="row g-2">

    <div className="col-md-6">
        <FormField
            label="Employee Name"
            name="annexureEmployeeName"

value={form.annexureEmployeeName}
            onChange={handleChange}
            placeholder="Employee name"
        />
    </div>

    <div className="col-md-6">
        <FormField
            label="Designation"
            name="annexureDesignation"
            value={form.annexureDesignation}
            onChange={handleChange}
            placeholder="Designation"
        />
    </div>

    <div className="col-md-6">
        <FormField
            label="Date of Joining"
            name="annexureDOJ"
            type="date"
            value={form.annexureDOJ}
            onChange={handleChange}

```

```

        />
    </div>

    <div className="col-md-6">
        <FormField
            label="CTC Amount (Rs.)"
            name="annexureCTC"
            type="number"
            value={form.annexureCTC}
            onChange={handleChange}
            placeholder="CTC"
        />
    </div>

    <div className="col-12">
        <FormField
            label="CTC Effective Date"
            name="annexureCTCEffectiveDate"
            type="date"
            value={
form.annexureCTCEffectiveDate
            }
            onChange={handleChange}
        />
    </div>

</div>

</div>

{ /*
===== */}
{ /* RIGHT COLUMN */}
{ /*
===== */}

<div className="col-lg-7">

    { /* COMPENSATION */}
    <SectionTitle
        number="04"
        title="Compensation Details"
        rightText="Monthly / Annual"
    />

    <div className="border rounded-3 overflow-

```



```
hidden mb-4">
```

```
bottom">
```

```
fw-semibold">
```

```
fw-semibold text-center">
```

```
fw-semibold text-center">
```

```
<div className="row g-0 bg-light border-
```

```
<div className="col-5">
```

```
<div className="px-3 py-2 small
```

```
Component
```

```
</div>
```

```
</div>
```

```
<div className="col-3">
```

```
<div className="px-2 py-2 small
```

```
Monthly
```

```
</div>
```

```
</div>
```

```
<div className="col-4">
```

```
<div className="px-2 py-2 small
```

```
Annual
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<CompensationRow
```

```
label="Basic"
```

```
monthly="basicMonthly"
```

```
annual="basicAnnual"
```

```
form={form}
```

```
onChange={handleChange}
```

```
/>
```

```
<CompensationRow
```

```
label="HRA"
```

```
monthly="hraMonthly"
```

```
annual="hraAnnual"
```

```
form={form}
```

```
onChange={handleChange}
```

```
/>
```

```
<CompensationRow
```

```
label="Special Allowance"
```

```
monthly="specialAllowanceMonthly"
```

```
annual="specialAllowanceAnnual"
```

```

        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Medical Expenses"
        monthly="medicalExpensesMonthly"
        annual="medicalExpensesAnnual"
        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Educational Allowance"
        monthly="educationalAllowanceMonthly"
        annual="educationalAllowanceAnnual"
        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Conveyance"
        monthly="conveyanceMonthly"
        annual="conveyanceAnnual"
        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Performance Incentive"
        monthly="performanceLinkMonthly"
        annual="performanceLinkAnnual"
        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Provident Fund"
        monthly="providentFundMonthly"
        annual="providentFundAnnual"
        form={form}
        onChange={handleChange}
    />

    <CompensationRow
        label="Gratuity"
        monthly="gratuityMonthly"

```

```

        annual="gratuityAnnual"
        form={form}
        onChange={handleChange}
    />

    <div className="border-top">
        <CompensationRow
            label="Total"
            monthly="totalMonthly"
            annual="totalAnnual"
            form={form}
            onChange={handleChange}
            strong
        />
    </div>

    <div className="bg-light">
        <CompensationRow
            label="Total Fixed CTC"
            monthly="totalFixedCTCMonthly"
            annual="totalFixedCTCAnnual"
            form={form}
            onChange={handleChange}
            strong
        />
    </div>

    <div className="row g-0 border-top">

        <div className="col-5">
            <div className="px-3 py-2 small
fw-semibold">

                Monthly In Hand
            </div>
        </div>

        <div className="col-7">
            <div className="px-2 py-1">
                <input
                    type="number"
                    className="form-control
form-control-sm"

                    name="monthlyInHand"
                    value={
                        form.monthlyInHand
                    }
                    onChange={handleChange}
                    placeholder="Amount"

```

```

        />
      </div>
    </div>

  </div>

  </div>

  { /* INVITEES */ }
  <SectionTitle
    number="07"
    title="Signing Parties"
    rightText="Up to 4 invitees"
  />

  <div className="row g-2">

    {[1, 2, 3, 4].map((index) => (
      <div
        className="col-md-6"
        key={index}
      >
        <InviteeCard
          index={index}
          form={form}
          handleChange={handleChange}
        />
      </div>
    ))}

  </div>

</div>

</div>

</div>

{ /* ===== */ }
*/}

{ /* ACTION BAR */ }
{ /* ===== */ }

*/}

<div className="card-footer bg-white border-top px-4
py-3">

```

```

align-items-center">
    <div className="small text-muted">
        <span className="text-danger">*</span>{" "}
        Required fields
    </div>

    <div className="d-flex gap-2">

        <button
            type="button"
            className="btn btn-light border px-4"
            onClick={() =>
                window.history.back()
            }
            disabled={loading}
        >
            Cancel
        </button>

        <button
            type="submit"
            className="btn btn-success px-4"
            disabled={loading}
        >
            {loading ? (
                <>
                    <span
                        className="spinner-border
spinner-border-sm me-2"
                        role="status"
                    />
                    Sending...
                </>
            ) : (
                <>
                    <span className="me-2">
                        ☒
                    </span>
                    Send Appointment Letter
                </>
            )}
        </button>

    </div>

```

```

        </div>

    </div>

</div>

</form>

</div>
</>
);
}

/* ===== */
/* SECTION TITLE */
/* ===== */

function SectionTitle({ number, title, rightText }) {
    return (
        <div className="d-flex align-items-center justify-content-between mb-2">

            <div className="d-flex align-items-center">

                <span
                    className="d-inline-flex align-items-center justify-content-
center rounded-circle bg-success-subtle text-success fw-semibold me-2"
                    style={{
                        width: "27px",
                        height: "27px",
                        fontSize: "11px",
                    }}
                >
                    {number}
                </span>

                <h6 className="mb-0 fw-semibold">
                    {title}
                </h6>

            </div>

            {rightText && (
                <small className="text-muted">
                    {rightText}
                </small>
            )}
        </div>
    );
}

```

```

        </div>
    );
}

/* ===== */
/* FORM FIELD */
/* ===== */

function FormField({
    label,
    name,
    value,
    onChange,
    type = "text",
    placeholder,
    required = false,
    textarea = false,
    rows = 2,
}) {
    return (
        <div>

            <label className="form-label small fw-semibold mb-1">
                {label}

                {required && (
                    <span className="text-danger ms-1">
                        *
                    </span>
                )}
            </label>

            {textarea ? (
                <textarea
                    rows={rows}
                    className="form-control form-control-sm"
                    name={name}
                    value={value}
                    onChange={onChange}
                    placeholder={placeholder}
                    required={required}
                />
            ) : (
                <input
                    type={type}
                    className="form-control form-control-sm"

```

```

        name={name}
        value={value}
        onChange={onChange}
        placeholder={placeholder}
        required={required}
    />
    )}

</div>
);
}

/* ===== */
/* COMPENSATION ROW */
/* ===== */

function CompensationRow({
    label,
    monthly,
    annual,
    form,
    onChange,
    strong = false,
}) {
    return (
        <div className="row g-0 border-bottom">

            <div className="col-5 d-flex align-items-center">

                <div
                    className={`px-3 py-1 small ${
                        strong ? "fw-semibold" : ""
                    }`}
                >
                    {label}
                </div>

            </div>

            <div className="col-3">

                <div className="px-1 py-1">

                    <input
                        type="number"
                        className={`form-control form-control-sm ${
                            strong ? "fw-semibold" : ""

```



```

        }`}
        name={monthly}
        value={form[monthly]}
        onChange={onChange}
        placeholder="0"
      />

    </div>

  </div>

  <div className="col-4">

    <div className="px-2 py-1">

      <input
        type="number"
        className={`form-control form-control-sm ${
          strong ? "fw-semibold" : ""
        }`}
        name={annual}
        value={form[annual]}
        onChange={onChange}
        placeholder="0"
      />

    </div>

  </div>

</div>
);
}

/* ===== */
/* INVITEE CARD */
/* ===== */

function InviteeCard({ index, form, handleChange }) {
  return (
    <div className="border rounded-3 p-3 bg-light h-100">

      <div className="d-flex align-items-center mb-2">

        <div
          className="d-flex align-items-center justify-content-center
rounded-circle bg-white border me-2"

```

```

        style={{
          width: "28px",
          height: "28px",
          fontSize: "12px",
          fontWeight: 600,
        }}
      >
        {index}
      </div>

    <div>
      <div className="fw-semibold small">
        Invitee {index}
      </div>

      <div
        className="text-muted"
        style={{ fontSize: "10px" }}
      >
        Signing party
      </div>
    </div>

  </div>

  <div className="mb-2">

    <input
      className="form-control form-control-sm"
      placeholder="Name"
      name={`invitee${index}Name`}
      value={form[`invitee${index}Name`]}
      onChange={handleChange}
    />

  </div>

  <div className="mb-2">

    <input
      type="email"
      className="form-control form-control-sm"
      placeholder="Email"
      name={`invitee${index}Email`}
      value={form[`invitee${index}Email`]}
      onChange={handleChange}
    />
  </div>

```

```
        </div>

        <input
          className="form-control form-control-sm"
          placeholder="Phone"
          name={`invitee${index}Phone`}
          value={form[`${invitee${index}Phone`}]}
          onChange={handleChange}
        />

      </div>
    );
  }
}
```