

```

import React, { useMemo, useState } from "react";
import { Card } from "@components/ui/card";
import { Button } from "@components/ui/button";
import { Input } from "@components/ui/input";
import { Textarea } from "@components/ui/textarea";
import { Separator } from "@components/ui/separator";
import { Check, Download, Sparkles, Copy, Printer } from "lucide-react";

/**
 * TalentPatriot — Dynamic Offer Letter Preview
 * -----
 * A single-file React component recruiters can use to preview an offer letter
 * before sending it to DocuSign/Dropbox Sign. Uses Tailwind + shadcn/ui.
 *
 * How to use:
 * 1) Drop this file into your app (e.g., src/components/OfferLetterPreview.tsx)
 * 2) Render <OfferLetterPreview org={...} candidate={...} job={...} />
 * 3) Hook the "Send for E-Sign" button to your backend action.
 */

export type Org = {
  name: string;
  logoUrl?: string;
  address?: string;
  phone?: string;
  email?: string;
  signatoryName?: string;
  signatoryTitle?: string;
  benefits?: string[];
};

export type Candidate = {
  firstName: string;
  lastName: string;
  address?: string;
};

export type Job = {
  title: string;
  startDate: string; // ISO or friendly date
  employmentType: "Full-Time" | "Part-Time" | "Contract";
  salaryAmount: string; // already formatted (e.g., "$120,000")
  salaryType: "per year" | "per hour" | "fixed";
  officeLocation?: string; // e.g., "Austin, TX" or "Remote"

```

```
    managerName?: string;
};
```

```
export type Props = {
  org?: Partial<Org>;
  candidate?: Partial<Candidate>;
  job?: Partial<Job>;
  today?: string; // override current date display
  onSend?: () => void; // wire up to your e-sign action
};
```

```
const defaultOrg: Org = {
  name: "Hildebrand Consulting Services",
  logoUrl: "https://dummyimage.com/160x48/1f3a5f/ffffff&text=TalentPatriot",
  address: "123 Main St, Springfield, USA",
  phone: "(555) 123-4567",
  email: "hello@example.com",
  signatoryName: "Alex Rivers",
  signatoryTitle: "Head of People",
  benefits: ["Health Insurance", "401(k)", "Paid Time Off"],
};
```

```
const defaultCandidate: Candidate = {
  firstName: "Jordan",
  lastName: "Lee",
  address: "45 Market Ave, Springfield, USA",
};
```

```
const defaultJob: Job = {
  title: "Software Engineer",
  startDate: new Date().toLocaleDateString(),
  employmentType: "Full-Time",
  salaryAmount: "$120,000",
  salaryType: "per year",
  officeLocation: "Remote",
  managerName: "Sam Patel",
};
```

```
export default function OfferLetterPreview({ org, candidate, job, today, onSend }: Props) {
  const [localOrg, setLocalOrg] = useState<Org>({ ...defaultOrg, ...org });
  const [localCandidate, setLocalCandidate] = useState<Candidate>({ ...defaultCandidate,
...candidate });
  const [localJob, setLocalJob] = useState<Job>({ ...defaultJob, ...job });
```

```

const [acceptBy, setAcceptBy] = useState<string>(new Date(Date.now() + 7 *
86400000).toLocaleDateString());

const dateDisplay = useMemo(() => today ?? new Date().toLocaleDateString(), [today]);

const fullName = `${localCandidate.firstName ?? ""} ${localCandidate.lastName ?? ""}`.trim();

const htmlToCopy = useMemo(() => {
  // Simple HTML snapshot of the letter for clipboard/e-sign template preview
  return `<!doctype html><meta charset="utf-8"/><div
style="font-family:Inter,system-ui,Arial,sans-serif;">` +
    `<div
style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px;">` +
    `<div
style="text-align:right;color:#334155;font-size:12px">${localOrg.name}<br/>${localOrg.address
?? ""}<br/>${localOrg.phone ?? ""} | ${localOrg.email ?? ""}</div></div>` +
    `<p style="color:#334155;font-size:14px">${dateDisplay}</p>` +
    `<p style="color:#334155;font-size:14px">${fullName}<br/>${localCandidate.address ??
""}</p>` +
    `<p style="color:#334155;font-size:14px">Dear ${localCandidate.firstName ?? ""},</p>` +
    `<p style="color:#334155;font-size:14px">We are pleased to offer you the position of
<b>${localJob.title}</b> with <b>${localOrg.name}</b>, reporting to ${localJob.managerName
?? "your hiring manager"} at our ${localJob.officeLocation ?? "office"} (or remote).</p>` +
    `<p style="color:#334155;font-size:14px">This position is ${localJob.employmentType} with
a start date of ${localJob.startDate}. Your compensation will be <b>${localJob.salaryAmount}
${localJob.salaryType}</b>, payable in accordance with our standard payroll practices.</p>` +
    `<p style="color:#334155;font-size:14px">Benefits:</p>` +
    `<ul>` + (localOrg.benefits ?? []).map(b => `<li>${b}</li>`).join("") + `</ul>` +
    `<p style="color:#334155;font-size:14px">This offer is contingent upon successful
completion of applicable background checks and I-9 verification. Please indicate your
acceptance by ${acceptBy}.</p>` +
    `<p style="color:#334155;font-size:14px">Sincerely,<br/><br/>${localOrg.signatoryName
?? ""}<br/>${localOrg.signatoryTitle ?? ""}<br/><b>${localOrg.name}</b></p>` +
    `</div>`;
  }, [localOrg, localCandidate, localJob, dateDisplay, acceptBy, fullName]);

const copyHtml = async () => {
  try { await navigator.clipboard.writeText(htmlToCopy); } catch {}
};

const printPage = () => window.print();

return (
  <div className="w-full p-4 md:p-6 grid grid-cols-1 lg:grid-cols-2 gap-6">

```

```

    {/ * Left: Controls */}
    <Card className="p-4 md:p-6 space-y-4">
      <div className="flex items-center gap-2">
        <Sparkles className="h-5 w-5" />
        <h2 className="text-lg font-semibold">Offer Letter Settings</h2>
      </div>
      <Separator />

      {/ * Org */}
      <h3 className="text-sm font-medium text-slate-600">Organization</h3>
      <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
        <Input value={localOrg.name} onChange={e=>setLocalOrg(o=>({...o,
name:e.target.value}})} placeholder="Org Name" />
        <Input value={localOrg.logoUrl ?? ""} onChange={e=>setLocalOrg(o=>({...o,
logoUrl:e.target.value}})} placeholder="Logo URL" />
        <Input value={localOrg.address ?? ""} onChange={e=>setLocalOrg(o=>({...o,
address:e.target.value}})} placeholder="Address" />
        <Input value={localOrg.phone ?? ""} onChange={e=>setLocalOrg(o=>({...o,
phone:e.target.value}})} placeholder="Phone" />
        <Input value={localOrg.email ?? ""} onChange={e=>setLocalOrg(o=>({...o,
email:e.target.value}})} placeholder="Email" />
        <Input value={localOrg.signatoryName ?? ""} onChange={e=>setLocalOrg(o=>({...o,
signatoryName:e.target.value}})} placeholder="Signatory Name" />
        <Input value={localOrg.signatoryTitle ?? ""} onChange={e=>setLocalOrg(o=>({...o,
signatoryTitle:e.target.value}})} placeholder="Signatory Title" />
        <Textarea value={localOrg.benefits ?? []}.join("\n")}
onChange={e=>setLocalOrg(o=>({...o, benefits:e.target.value.split(/\n+/).filter(Boolean)}))}
placeholder={"Benefits (one per line)"} />
      </div>

      {/ * Candidate */}
      <h3 className="text-sm font-medium text-slate-600 mt-4">Candidate</h3>
      <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
        <Input value={localCandidate.firstName} onChange={e=>setLocalCandidate(c=>({...c,
firstName:e.target.value}})} placeholder="First name" />
        <Input value={localCandidate.lastName} onChange={e=>setLocalCandidate(c=>({...c,
lastName:e.target.value}})} placeholder="Last name" />
        <Input className="md:col-span-2" value={localCandidate.address ?? ""}
onChange={e=>setLocalCandidate(c=>({...c, address:e.target.value}})} placeholder="Address"
/>
      </div>

      {/ * Job */}
      <h3 className="text-sm font-medium text-slate-600 mt-4">Job</h3>

```

```

    <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
      <Input value={localJob.title} onChange={e=>setLocalJob(j=>({...j, title:e.target.value}))}
placeholder="Job title" />
      <Input value={localJob.startDate} onChange={e=>setLocalJob(j=>({...j,
startDate:e.target.value}))} placeholder="Start date" />
      <Input value={localJob.salaryAmount} onChange={e=>setLocalJob(j=>({...j,
salaryAmount:e.target.value}))} placeholder="Salary amount (e.g., $120,000)" />
      <Input value={localJob.salaryType} onChange={e=>setLocalJob(j=>({...j,
salaryType:e.target.value as Job["salaryType"]}))} placeholder="Salary type (per
year/hour/fixed)" />
      <Input value={localJob.employmentType} onChange={e=>setLocalJob(j=>({...j,
employmentType:e.target.value as Job["employmentType"]}))} placeholder="Employment type"
/>
      <Input value={localJob.officeLocation ?? ""} onChange={e=>setLocalJob(j=>({...j,
officeLocation:e.target.value}))} placeholder="Office location or Remote" />
      <Input value={localJob.managerName ?? ""} onChange={e=>setLocalJob(j=>({...j,
managerName:e.target.value}))} placeholder="Hiring manager" />
    </div>

```

```

    <div className="grid grid-cols-1 md:grid-cols-2 gap-3 mt-3">
      <Input value={acceptBy} onChange={e=>setAcceptBy(e.target.value)}
placeholder="Accept by date" />
    </div>

```

```

    <div className="flex gap-2 pt-4">
      <Button onClick={printPage} variant="secondary"><Printer className="h-4 w-4
mr-2"/>Print</Button>
      <Button onClick={copyHtml} variant="outline"><Copy className="h-4 w-4 mr-2"/>Copy
HTML</Button>
      <Button onClick={onSend} className="ml-auto"><Check className="h-4 w-4
mr-2"/>Send for E-Sign</Button>
    </div>
  </Card>

```

```

  {/* Right: Preview */}
  <Card className="p-6 bg-white shadow-sm">
    <div className="max-w-[720px] mx-auto print:max-w-none">
      {/* Header */}
      <div className="flex items-start justify-between">
        {localOrg.logoUrl ? (
          // eslint-disable-next-line @next/next/no-img-element
          <img src={localOrg.logoUrl} alt="logo" className="h-10 object-contain" />
        ) : (
          <div className="text-xl font-semibold text-slate-800">{localOrg.name}</div>

```

```

    })
    <div className="text-right text-xs text-slate-500">
      <div>{localOrg.name}</div>
      {localOrg.address && <div>{localOrg.address}</div>}
      {(localOrg.phone || localOrg.email) && (
        <div>[{localOrg.phone, localOrg.email}.filter(Boolean).join(" · ")]</div>
      )}
    </div>
  </div>

  <Separator className="my-4" />

  {/* Meta */}
  <div className="text-sm text-slate-600">{dateDisplay}</div>
  <div className="mt-2 text-sm text-slate-700">
    <div>{fullName}</div>
    {localCandidate.address && <div>{localCandidate.address}</div>}
  </div>

  <div className="mt-6 text-slate-800 leading-7">
    <p>Dear {localCandidate.firstName},</p>
    <p className="mt-4">
      We are pleased to offer you the position of <b>{localJob.title}</b> with
      <b>{localOrg.name}</b>,
      reporting to {localJob.managerName ?? "your hiring manager"} at our
      {localJob.officeLocation ?? "office"} (or remote).
    </p>
    <p className="mt-4">
      This position is {localJob.employmentType} with a start date of {localJob.startDate}.
      Your
      compensation will be <b>{localJob.salaryAmount} {localJob.salaryType}</b>, payable
      in
      accordance with our standard payroll practices.
    </p>
  </div>

  {/* Comp card */}
  <div className="mt-6 grid grid-cols-3 rounded-2xl border overflow-hidden">
    <div className="bg-[#1F3A5F] text-white p-3 text-center text-sm
font-semibold">Compensation</div>
    <div className="bg-[#1F3A5F] text-white p-3 text-center text-sm font-semibold">Start
Date</div>
    <div className="bg-[#1F3A5F] text-white p-3 text-center text-sm
font-semibold">Status</div>

```

```
<div className="col-span-1 p-3 text-center text-slate-700">{localJob.salaryAmount}
{localJob.salaryType}</div>
<div className="col-span-1 p-3 text-center text-slate-700">{localJob.startDate}</div>
<div className="col-span-1 p-3 text-center
text-slate-700">{localJob.employmentType}</div>
</div>
```

```
{/* Benefits */}
{localOrg.benefits && localOrg.benefits.length > 0 && (
  <div className="mt-6">
    <div className="text-sm font-semibold text-slate-700">Benefits</div>
    <ul className="mt-2 list-disc list-inside text-slate-700 space-y-1">
      {localOrg.benefits.map((b, i) => (
        <li key={i}>{b}</li>
      ))}
    </ul>
  </div>
)}
```

```
<div className="mt-6 text-slate-800 leading-7">
  <p>
    This offer is contingent upon successful completion of applicable background checks
    and I-9 verification.
    Please indicate your acceptance by {acceptBy} by signing below or via our e-signature
    system.
  </p>
  <p className="mt-4">We are excited to welcome you to <b>{localOrg.name}</b> and
  look forward to your contributions.</p>
</div>
```

```
{/* Signature */}
<div className="mt-8">
  <div className="text-slate-800">Sincerely,</div>
  <div className="mt-6">
    <div className="font-medium text-slate-900">{localOrg.signatoryName}</div>
    <div className="text-slate-600">{localOrg.signatoryTitle}</div>
    <div className="font-semibold text-slate-900">{localOrg.name}</div>
  </div>
```

```
<div className="mt-10 pt-6 border-t text-slate-700">
  <div className="text-sm">Accepted by:</div>
  <div className="mt-6 h-8 border-b w-72" />
  <div className="mt-4 text-sm">Date:</div>
  <div className="mt-2 h-8 border-b w-40" />
```

```
        </div>
      </div>
    </div>
  </Card>
</div>
);
}
```