



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# MARCH 2025

**EPAM - 22CS2241F**

**PROJECT REPORT**

**ONLINE RESUME BUILDER**



TEAM EPAM

K L UNIVERSITY | HYDERABAD



|  |  |
| --- | --- |
| **STUDENT NAME** | V.Thanmayee I. Kavya Priya V. Thanmai Naga Siri  T. Taniswi |
| **STUDENT ID** | 2210030072 2210030071 2210030163 2210030402 |
| **YEAR** | 3Rd |
| **SEMESTER** | EVEN SEM |
| **SECTION** | AI-ML-A2 |
| **FACULTY NAME** | Dr. Pavan Kumar Pagadala |

ONLINE RESUME BUILDER

# Introduction:

An online resume builder is a web-based tool that enables users to create professional resumes without needing advanced design or formatting skills. These platforms offer pre-built templates, easy-to-use interfaces, and customization options to help job seekers craft resumes efficiently.

**Key Features of E-Learning Platforms:**

* **Template Selection:** Professionally designed resume templates.
* **Form-Based Data Entry:** Step-by-step guided input for education, experience, skills, etc.
* **Real-Time Preview:** See how the resume looks while editing.
* **Export Options:** Download as PDF, DOCX, or share via link.
* **Customization Tools:** Fonts, color schemes, section reordering.

**Benefits:**

* Time-saving and user-friendly.
* No design or technical skills required.
* Professional and clean output.
* Accessible anytime from any device.

**Challenges:**

* Ensuring templates are ATS-compatible.
* Maintaining responsiveness across devices.
* Handling user data privacy and security.
* Avoiding plagiarism or generic content.

**Future Scope:**

* Integration with job portals (e.g., LinkedIn, Indeed).
* AI-based resume scoring and optimization.
* Multilingual support.
* Video resume or portfolio integration.

**User Flow / Workflow:**

1. Sign Up / Log In
2. Choose a Template
3. Enter Personal Details
4. Add Work Experience, Education, Skills, etc.
5. Customize Layout and Design
6. Preview and Download Resume

**Modules of the System:**

* **Authentication Module**: Handles user sign-up/login.
* **Template Module**: Stores and displays available resume templates.
* **Editor Module**: Input and edit content.
* **Preview Module**: Real-time rendering of resume.
* **Export Module**: Converts data into downloadable formats.
* **Database Module**: Stores user data securely.

**Conclusion:**

The Online Resume Builder is a practical and user-centric solution aimed at simplifying the resume creation process for users of all backgrounds. By providing customizable templates, intuitive interfaces, and export-ready formats, the platform empowers individuals to present their professional information in a polished and effective manner.  
  
  
**Code:**

**Sign in**  
import { SignIn } from "@clerk/nextjs";

import React from "react";

const Page = () => {

return (

<div className="flex h-screen items-center justify-center flex-col p-10">

<SignIn forceRedirectUrl={"/dashboard"} routing="hash" />

</div>

);

};

export default Page;

**Sign up**

import { SignUp } from "@clerk/nextjs";

import React from "react";

const Page = () => {

return (

<div className="flex h-screen items-center justify-center flex-col p-10">

<SignUp forceRedirectUrl={"/dashboard"} routing="hash" />

</div>

);

};

export default Page;

**Input**

import \* as React from "react"

import { cn } from "@/lib/utils"

export interface InputProps

extends React.InputHTMLAttributes<HTMLInputElement> {}

const Input = React.forwardRef<HTMLInputElement, InputProps>(

({ className, type, ...props }, ref) => {

return (

<input

type={type}

className={cn(

"flex h-10 w-full rounded-md border border-slate-200 bg-white px-3 py-2 text-sm ring-offset-white file:border-0 file:bg-transparent file:text-sm file:font-medium placeholder:text-slate-500 focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-slate-950 focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50 dark:border-slate-800 dark:bg-slate-950 dark:ring-offset-slate-950 dark:placeholder:text-slate-400 dark:focus-visible:ring-slate-300",

className

)}

ref={ref}

{...props}

/>

)

}

)

Input.displayName = "Input"

export { Input }

**Form**

"use client"

import \* as React from "react"

import \* as LabelPrimitive from "@radix-ui/react-label"

import { Slot } from "@radix-ui/react-slot"

import {

Controller,

ControllerProps,

FieldPath,

FieldValues,

FormProvider,

useFormContext,

} from "react-hook-form"

import { cn } from "@/lib/utils"

import { Label } from "@/components/ui/label"

const Form = FormProvider

type FormFieldContextValue<

TFieldValues extends FieldValues = FieldValues,

TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>

> = {

name: TName

}

const FormFieldContext = React.createContext<FormFieldContextValue>(

{} as FormFieldContextValue

)

const FormField = <

TFieldValues extends FieldValues = FieldValues,

TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>

>({

...props

}: ControllerProps<TFieldValues, TName>) => {

return (

<FormFieldContext.Provider value={{ name: props.name }}>

<Controller {...props} />

</FormFieldContext.Provider>

)

}

const useFormField = () => {

const fieldContext = React.useContext(FormFieldContext)

const itemContext = React.useContext(FormItemContext)

const { getFieldState, formState } = useFormContext()

const fieldState = getFieldState(fieldContext.name, formState)

if (!fieldContext) {

throw new Error("useFormField should be used within <FormField>")

}

const { id } = itemContext

return {

id,

name: fieldContext.name,

formItemId: `${id}-form-item`,

formDescriptionId: `${id}-form-item-description`,

formMessageId: ${id}-form-item-message,

...fieldState,

}

}

type FormItemContextValue = {

id: string

}

const FormItemContext = React.createContext<FormItemContextValue>(

{} as FormItemContextValue

)

const FormItem = React.forwardRef<

HTMLDivElement,

React.HTMLAttributes<HTMLDivElement>

>(({ className, ...props }, ref) => {

const id = React.useId()

return (

<FormItemContext.Provider value={{ id }}>

<div ref={ref} className={cn("space-y-2", className)} {...props} />

</FormItemContext.Provider>

)

})

FormItem.displayName = "FormItem"

const FormLabel = React.forwardRef<

React.ElementRef<typeof LabelPrimitive.Root>,

React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root>

>(({ className, ...props }, ref) => {

const { error, formItemId } = useFormField()

return (

<Label

ref={ref}

className={cn(error && "text-red-500 dark:text-red-900", className)}

htmlFor={formItemId}

{...props}

/>

)

})

FormLabel.displayName = "FormLabel"

const FormControl = React.forwardRef<

React.ElementRef<typeof Slot>,

React.ComponentPropsWithoutRef<typeof Slot>

>(({ ...props }, ref) => {

const { error, formItemId, formDescriptionId, formMessageId } = useFormField()

return (

<Slot

ref={ref}

id={formItemId}

aria-describedby={

!error

? ${formDescriptionId}

: ${formDescriptionId} ${formMessageId}

}

aria-invalid={!!error}

{...props}

/>

)

})

FormControl.displayName = "FormControl"

const FormDescription = React.forwardRef<

HTMLParagraphElement,

React.HTMLAttributes<HTMLParagraphElement>

>(({ className, ...props }, ref) => {

const { formDescriptionId } = useFormField()

return (

<p

ref={ref}

id={formDescriptionId}

className={cn("text-sm text-slate-500 dark:text-slate-400", className)}

{...props}

/>

)

})

FormDescription.displayName = "FormDescription"

const FormMessage = React.forwardRef<

HTMLParagraphElement,

React.HTMLAttributes<HTMLParagraphElement>

>(({ className, children, ...props }, ref) => {

const { error, formMessageId } = useFormField()

const body = error ? String(error?.message) : children

if (!body) {

return null

}

return (

<p

ref={ref}

id={formMessageId}

className={cn("text-sm font-medium text-red-500 dark:text-red-900", className)}

{...props}

>

{body}

</p>

)

})

FormMessage.displayName = "FormMessage"

export {

useFormField,

Form,

FormItem,

FormLabel,

FormControl,

FormDescription,

FormMessage,

FormField,

}

**Label**

"use client"

import \* as React from "react"

import \* as LabelPrimitive from "@radix-ui/react-label"

import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "@/lib/utils"

const labelVariants = cva(

"text-sm font-medium leading-none peer-disabled:cursor-not-allowed peer-disabled:opacity-70"

)

const Label = React.forwardRef<

React.ElementRef<typeof LabelPrimitive.Root>,

React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root> &

VariantProps<typeof labelVariants>

>(({ className, ...props }, ref) => (

<LabelPrimitive.Root

ref={ref}

className={cn(labelVariants(), className)}

{...props}

/>

))

Label.displayName = LabelPrimitive.Root.displayName

export { Label }

**Button**

import \* as React from "react";

import { Slot } from "@radix-ui/react-slot";

import { cva, type VariantProps } from "class-variance-authority";

import { cn } from "@/lib/utils";

const buttonVariants = cva(

"inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm font-medium ring-offset-white transition-colors focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-slate-950 focus-visible:ring-offset-2 disabled:pointer-events-none disabled:opacity-50 dark:ring-offset-slate-950 dark:focus-visible:ring-slate-300",

{

variants: {

variant: {

default:

"bg-slate-900 text-slate-50 hover:bg-slate-900/90 dark:bg-slate-50 dark:text-slate-900 dark:hover:bg-slate-50/90",

destructive:

"bg-red-500 text-slate-50 hover:bg-red-500/90 dark:bg-red-900 dark:text-slate-50 dark:hover:bg-red-900/90",

outline:

"border border-slate-200 bg-white hover:bg-slate-100 hover:text-slate-900 dark:border-slate-800 dark:bg-slate-950 dark:hover:bg-slate-800 dark:hover:text-slate-50",

secondary:

"bg-slate-100 text-slate-900 hover:bg-slate-100/80 dark:bg-slate-800 dark:text-slate-50 dark:hover:bg-slate-800/80",

ghost:

"hover:bg-slate-100 hover:text-slate-900 dark:hover:bg-slate-800 dark:hover:text-slate-50",

link: "text-slate-900 underline-offset-4 hover:underline dark:text-slate-50",

},

size: {

default: "h-10 px-4 py-2",

sm: "h-9 rounded-md px-3",

lg: "h-11 rounded-md px-8",

icon: "h-10 w-10",

},

},

defaultVariants: {

variant: "default",

size: "default",

},

}

);

export interface ButtonProps

extends React.ButtonHTMLAttributes<HTMLButtonElement>,

VariantProps<typeof buttonVariants> {

asChild?: boolean;

}

const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(

({ className, variant, size, asChild = false, ...props }, ref) => {

const Comp = asChild ? Slot : "button";

return (

<Comp

className={cn(buttonVariants({ variant, size, className }))}

ref={ref}

{...props}

/>

);

}

);

Button.displayName = "Button";

export { Button, buttonVariants };

**Dialog**

"use client"

import \* as React from "react"

import \* as DialogPrimitive from "@radix-ui/react-dialog"

import { X } from "lucide-react"

import { cn } from "@/lib/utils"

const Dialog = DialogPrimitive.Root

const DialogTrigger = DialogPrimitive.Trigger

const DialogPortal = DialogPrimitive.Portal

const DialogClose = DialogPrimitive.Close

const DialogOverlay = React.forwardRef<

React.ElementRef<typeof DialogPrimitive.Overlay>,

React.ComponentPropsWithoutRef<typeof DialogPrimitive.Overlay>

>(({ className, ...props }, ref) => (

<DialogPrimitive.Overlay

ref={ref}

className={cn(

"fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",

className

)}

{...props}

/>

))

DialogOverlay.displayName = DialogPrimitive.Overlay.displayName

const DialogContent = React.forwardRef<

React.ElementRef<typeof DialogPrimitive.Content>,

React.ComponentPropsWithoutRef<typeof DialogPrimitive.Content>

>(({ className, children, ...props }, ref) => (

<DialogPortal>

<DialogOverlay />

<DialogPrimitive.Content

ref={ref}

className={cn(

"fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 border border-slate-200 bg-white p-6 shadow-lg duration-200 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-[state=closed]:slide-out-to-left-1/2 data-[state=closed]:slide-out-to-top-[48%] data-[state=open]:slide-in-from-left-1/2 data-[state=open]:slide-in-from-top-[48%] sm:rounded-lg dark:border-slate-800 dark:bg-slate-950",

className

)}

{...props}

>

{children}

<DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-white transition-opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-slate-950 focus:ring-offset-2 disabled:pointer-events-none data-[state=open]:bg-slate-100 data-[state=open]:text-slate-500 dark:ring-offset-slate-950 dark:focus:ring-slate-300 dark:data-[state=open]:bg-slate-800 dark:data-[state=open]:text-slate-400">

<X className="h-4 w-4" />

<span className="sr-only">Close</span>

</DialogPrimitive.Close>

</DialogPrimitive.Content>

</DialogPortal>

))

DialogContent.displayName = DialogPrimitive.Content.displayName

const DialogHeader = ({

className,

...props

}: React.HTMLAttributes<HTMLDivElement>) => (

<div

className={cn(

"flex flex-col space-y-1.5 text-center sm:text-left",

className

)}

{...props}

/>

)

DialogHeader.displayName = "DialogHeader"

const DialogFooter = ({

className,

...props

}: React.HTMLAttributes<HTMLDivElement>) => (

<div

className={cn(

"flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",

className

)}

{...props}

/>

)

DialogFooter.displayName = "DialogFooter"

const DialogTitle = React.forwardRef<

React.ElementRef<typeof DialogPrimitive.Title>,

React.ComponentPropsWithoutRef<typeof DialogPrimitive.Title>

>(({ className, ...props }, ref) => (

<DialogPrimitive.Title

ref={ref}

className={cn(

"text-lg font-semibold leading-none tracking-tight",

className

)}

{...props}

/>

))

DialogTitle.displayName = DialogPrimitive.Title.displayName

const DialogDescription = React.forwardRef<

React.ElementRef<typeof DialogPrimitive.Description>,

React.ComponentPropsWithoutRef<typeof DialogPrimitive.Description>

>(({ className, ...props }, ref) => (

<DialogPrimitive.Description

ref={ref}

className={cn("text-sm text-slate-500 dark:text-slate-400", className)}

{...props}

/>

))

DialogDescription.displayName = DialogPrimitive.Description.displayName

export {

Dialog,

DialogPortal,

DialogOverlay,

DialogClose,

DialogTrigger,

DialogContent,

DialogHeader,

DialogFooter,

DialogTitle,

DialogDescription,

}

**Drop Down**

"use client"

import \* as React from "react"

import \* as DropdownMenuPrimitive from "@radix-ui/react-dropdown-menu"

import { Check, ChevronRight, Circle } from "lucide-react"

import { cn } from "@/lib/utils"

const DropdownMenu = DropdownMenuPrimitive.Root

const DropdownMenuTrigger = DropdownMenuPrimitive.Trigger

const DropdownMenuGroup = DropdownMenuPrimitive.Group

const DropdownMenuPortal = DropdownMenuPrimitive.Portal

const DropdownMenuSub = DropdownMenuPrimitive.Sub

const DropdownMenuRadioGroup = DropdownMenuPrimitive.RadioGroup

const DropdownMenuSubTrigger = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.SubTrigger>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubTrigger> & {

inset?: boolean

}

>(({ className, inset, children, ...props }, ref) => (

<DropdownMenuPrimitive.SubTrigger

ref={ref}

className={cn(

"flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-slate-100 data-[state=open]:bg-slate-100 dark:focus:bg-slate-800 dark:data-[state=open]:bg-slate-800",

inset && "pl-8",

className

)}

{...props}

>

{children}

<ChevronRight className="ml-auto h-4 w-4" />

</DropdownMenuPrimitive.SubTrigger>

))

DropdownMenuSubTrigger.displayName =

DropdownMenuPrimitive.SubTrigger.displayName

const DropdownMenuSubContent = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.SubContent>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubContent>

>(({ className, ...props }, ref) => (

<DropdownMenuPrimitive.SubContent

ref={ref}

className={cn(

"z-50 min-w-[8rem] overflow-hidden rounded-md border border-slate-200 bg-white p-1 text-slate-950 shadow-lg data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 dark:border-slate-800 dark:bg-slate-950 dark:text-slate-50",

className

)}

{...props}

/>

))

DropdownMenuSubContent.displayName =

DropdownMenuPrimitive.SubContent.displayName

const DropdownMenuContent = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.Content>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Content>

>(({ className, sideOffset = 4, ...props }, ref) => (

<DropdownMenuPrimitive.Portal>

<DropdownMenuPrimitive.Content

ref={ref}

sideOffset={sideOffset}

className={cn(

"z-50 min-w-[8rem] overflow-hidden rounded-md border border-slate-200 bg-white p-1 text-slate-950 shadow-md data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 dark:border-slate-800 dark:bg-slate-950 dark:text-slate-50",

className

)}

{...props}

/>

</DropdownMenuPrimitive.Portal>

))

DropdownMenuContent.displayName = DropdownMenuPrimitive.Content.displayName

const DropdownMenuItem = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.Item>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Item> & {

inset?: boolean

}

>(({ className, inset, ...props }, ref) => (

<DropdownMenuPrimitive.Item

ref={ref}

className={cn(

"relative flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none transition-colors focus:bg-slate-100 focus:text-slate-900 data-[disabled]:pointer-events-none data-[disabled]:opacity-50 dark:focus:bg-slate-800 dark:focus:text-slate-50",

inset && "pl-8",

className

)}

{...props}

/>

))

DropdownMenuItem.displayName = DropdownMenuPrimitive.Item.displayName

const DropdownMenuCheckboxItem = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.CheckboxItem>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.CheckboxItem>

>(({ className, children, checked, ...props }, ref) => (

<DropdownMenuPrimitive.CheckboxItem

ref={ref}

className={cn(

"relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none transition-colors focus:bg-slate-100 focus:text-slate-900 data-[disabled]:pointer-events-none data-[disabled]:opacity-50 dark:focus:bg-slate-800 dark:focus:text-slate-50",

className

)}

checked={checked}

{...props}

>

<span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">

<DropdownMenuPrimitive.ItemIndicator>

<Check className="h-4 w-4" />

</DropdownMenuPrimitive.ItemIndicator>

</span>

{children}

</DropdownMenuPrimitive.CheckboxItem>

))

DropdownMenuCheckboxItem.displayName =

DropdownMenuPrimitive.CheckboxItem.displayName

const DropdownMenuRadioItem = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.RadioItem>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.RadioItem>

>(({ className, children, ...props }, ref) => (

<DropdownMenuPrimitive.RadioItem

ref={ref}

className={cn(

"relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none transition-colors focus:bg-slate-100 focus:text-slate-900 data-[disabled]:pointer-events-none data-[disabled]:opacity-50 dark:focus:bg-slate-800 dark:focus:text-slate-50",

className

)}

{...props}

>

<span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">

<DropdownMenuPrimitive.ItemIndicator>

<Circle className="h-2 w-2 fill-current" />

</DropdownMenuPrimitive.ItemIndicator>

</span>

{children}

</DropdownMenuPrimitive.RadioItem>

))

DropdownMenuRadioItem.displayName = DropdownMenuPrimitive.RadioItem.displayName

const DropdownMenuLabel = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.Label>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label> & {

inset?: boolean

}

>(({ className, inset, ...props }, ref) => (

<DropdownMenuPrimitive.Label

ref={ref}

className={cn(

"px-2 py-1.5 text-sm font-semibold",

inset && "pl-8",

className

)}

{...props}

/>

))

DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName

const DropdownMenuSeparator = React.forwardRef<

React.ElementRef<typeof DropdownMenuPrimitive.Separator>,

React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Separator>

>(({ className, ...props }, ref) => (

<DropdownMenuPrimitive.Separator

ref={ref}

className={cn("-mx-1 my-1 h-px bg-slate-100 dark:bg-slate-800", className)}

{...props}

/>

))

DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName

const DropdownMenuShortcut = ({

className,

...props

}: React.HTMLAttributes<HTMLSpanElement>) => {

return (

<span

className={cn("ml-auto text-xs tracking-widest opacity-60", className)}

{...props}

/>

)

}

DropdownMenuShortcut.displayName = "DropdownMenuShortcut"

export {

DropdownMenu,

DropdownMenuTrigger,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuCheckboxItem,

DropdownMenuRadioItem,

DropdownMenuLabel,

DropdownMenuSeparator,

DropdownMenuShortcut,

DropdownMenuGroup,

DropdownMenuPortal,

DropdownMenuSub,

DropdownMenuSubContent,

DropdownMenuSubTrigger,

DropdownMenuRadioGroup,

}

**Header**

"use client";

import { UserButton, useUser } from "@clerk/nextjs";

import Link from "next/link";

import React from "react";

const Header = () => {

const user = useUser();

return (

<header className="sticky top-0 z-50">

<nav className="backdrop-blur-md px-6 py-2.5">

<div className="flex flex-wrap justify-between items-center mx-auto max-w-screen-xl">

<Link href="/" className="flex items-center">

<img src="/icons/logo.svg" className="mr-3 h-7 sm:h-9" alt="logo" />

<span className="self-center text-xl font-bold whitespace-nowrap">

ResumeAI

</span>

</Link>

<div className="flex items-center lg:order-2">

{user?.isLoaded && !user?.isSignedIn ? (

<Link

href="/sign-in"

className="text-gray-800 hover:bg-primary-700/10 duration-300 focus:ring-4 focus:ring-primary-700/30 font-medium rounded-full text-sm px-4 lg:px-5 py-2 lg:py-2.5 mr-2 focus:outline-none"

>

Log in

</Link>

) : (

<>

<div className="mr-4 h-full items-center align-middle flex max-md:hidden justify-center">

<UserButton showName={true} />

</div>

<div className="mr-4 h-full items-center align-middle hidden max-md:flex justify-center">

<UserButton showName={false} />

</div>

</>

)}

<Link

href={`${!user?.isSignedIn ? "/sign-up" : "/dashboard"}`}

className="text-white bg-primary-700 hover:bg-primary-800 focus:ring-4 focus:ring-primary-300 font-medium rounded-full text-sm px-4 lg:px-5 py-2 lg:py-2.5 focus:outline-none"

>

{!user?.isSignedIn ? "Get started" : "Dashboard"}

</Link>

</div>

</div>

</nav>

</header>

);

};

export default Header;

**Header Text**  
import React from "react";

const HeaderText = ({ title, subtitle }: { title: string; subtitle: string }) => {

return (

<>

<h2 className="font-bold text-3xl">{title}</h2>

<p className="text-slate-600">{subtitle}</p>

</>

);

};

export default HeaderText;

**Resume View**  
"use client";

import Header from "@/components/layout/Header";

import { Button } from "@/components/ui/button";

import { FormProvider } from "@/lib/context/FormProvider";

import { RWebShare } from "react-web-share";

import React, { useEffect, useState } from "react";

import ResumePreview from "@/components/layout/my-resume/ResumePreview";

import { usePathname } from "next/navigation";

import PageWrapper from "@/components/common/PageWrapper";

import { DownloadIcon, Share2Icon } from "lucide-react";

import { fetchResume } from "@/lib/actions/resume.actions";

import html2pdf from "html2pdf.js";

interface FinalResumeViewProps {

params: { id: string };

isOwnerView: boolean;

}

const FinalResumeView: React.FC<FinalResumeViewProps> = ({

params,

isOwnerView,

}) => {

const [download, setDownload] = useState<boolean>(false);

const path = usePathname();

const [formData, setFormData] = useState<any>({});

useEffect(() => {

const loadResumeData = async () => {

try {

const resumeData = await fetchResume(params.id);

setFormData(JSON.parse(resumeData));

} catch (error) {

console.error("Error fetching resume:", error);

}

};

loadResumeData();

}, [params.id]);

const sanitize = (str: string | undefined | null): string =>

str?.trim().replace(/\s+/g, "\_") || "User\_Resume";

const handleDownloadPDF = () => {

const element = document.getElementById("print-area");

const opt = {

margin: 0,

filename: `${sanitize(

`${formData?.firstName ?? "User"}\_${formData?.lastName ?? ""}\_${

formData?.jobTitle ?? ""

}\_Resume.pdf`

)}`,

image: { type: "jpeg", quality: 0.98 },

html2canvas: { scale: 2 },

jsPDF: { unit: "in", format: "a4", orientation: "portrait" },

};

if (element) {

setDownload(true);

html2pdf()

.set(opt)

.from(element)

.save()

.finally(() => {

setDownload(false);

});

}

};

return (

<>

<PageWrapper>

<FormProvider params={params}>

<div id="no-print">

<Header />

<div className="my-10 mx-10 md:mx-20 lg:mx-36">

{isOwnerView ? (

<>

<h2 className="text-center text-2xl font-bold">

Congrats! Your ultimate AI-generated resume is ready!

</h2>

<p className="text-center text-gray-600">

You can now download your resume or share its unique URL

with your friends and family.

</p>

</>

) : (

<>

<h2 className="text-center text-2xl font-bold">

Resume Preview

</h2>

<p className="text-center text-gray-600">

You are currently viewing a preview of someone else's

resume.

</p>

</>

)}

<div className="flex max-sm:flex-col justify-center gap-8 my-10">

<Button

className="flex px-12 py-6 gap-2 rounded-full bg-primary-700 hover:bg-primary-800 focus:ring-4 focus:ring-primary-700/30 text-white"

onClick={() => handleDownloadPDF()}

>

<DownloadIcon className="size-6" /> Download

</Button>

<RWebShare

data={{

text: "Check out my resume!",

url: `${path}`,

title: `${formData?.firstName ?? "User"} ${

formData?.lastName ?? "Resume"

}'s Resume`,

}}

onClick={() => console.log("Shared successfully!")}

>

<Button className="flex px-12 py-6 gap-2 rounded-full bg-slate-200 hover:bg-primary-700/20 focus:ring-4 focus:ring-primary-700/30 text-black">

<Share2Icon className="size-6" /> Share URL

</Button>

</RWebShare>

</div>

</div>

</div>

<div className="px-10 pt-4 pb-16 max-sm:px-5 max-sm:pb-8 print:p-0">

<div id="print-area">

<ResumePreview download={download} />

</div>

</div>

</FormProvider>

</PageWrapper>

</>

);

};

export default FinalResumeView;

**Mebsar**  
"use client";

import { useFormContext } from "@/lib/context/FormProvider";

import { useEffect, useState } from "react";

import {

Popover,

PopoverContent,

PopoverTrigger,

} from "@/components/ui/popover";

import { Button } from "../ui/button";

import { Check, LayoutGrid } from "lucide-react";

import { themeColors } from "@/lib/utils";

import { updateResume } from "@/lib/actions/resume.actions";

import { useToast } from "../ui/use-toast";

const ThemeColor = ({ params }: { params: { id: string } }) => {

const { toast } = useToast();

const { formData, handleInputChange } = useFormContext();

const [selectedColor, setSelectedColor] = useState(themeColors[0]);

useEffect(() => {

setSelectedColor(formData.themeColor);

}, [formData.themeColor]);

const onColorSelect = async (color: any) => {

setSelectedColor(color);

handleInputChange({

target: {

name: "themeColor",

value: color,

},

});

const result = await updateResume({

resumeId: params.id,

updates: {

themeColor: color,

},

});

if (result.success) {

toast({

title: "Information saved.",

description: "Theme color updated successfully.",

className: "bg-white",

});

} else {

toast({

title: "Uh Oh! Something went wrong.",

description: result?.error,

variant: "destructive",

className: "bg-white",

});

}

};

return (

<Popover>

<PopoverTrigger asChild>

<Button size="sm" className="btn-gradient flex gap-2">

{" "}

<LayoutGrid /> Theme

</Button>

</PopoverTrigger>

<PopoverContent>

<h2 className="mb-3 text-sm font-bold">Select Theme Color</h2>

<div className="grid grid-cols-5 gap-3">

{themeColors.map((item, index) => (

<div

key={index}

onClick={() => onColorSelect(item)}

className="flex justify-center items-center h-8 w-8 rounded-lg cursor-pointer hover:scale-110 transition-all duration-300"

style={{

background: item,

}}

>

{selectedColor == item && (

<Check color="#ffffff" strokeWidth={3} width={20} height={20} />

)}

</div>

))}

</div>

</PopoverContent>

</Popover>

);

};

export default ThemeColor;

**Layout**

import PageWrapper from "@/components/common/PageWrapper";

import React from "react";

export default function RootLayout({

children,

}: Readonly<{

children: React.ReactNode;

}>) {

return (

<main>

<PageWrapper>{children}</PageWrapper>

</main>

);

}

Page

"use client";

import Header from "@/components/layout/Header";

import { useUser } from "@clerk/nextjs";

import { ArrowBigUp, AtomIcon, Edit, Share2 } from "lucide-react";

import Link from "next/link";

import React from "react";

const page = () => {

const user = useUser();

return (

<div>

<Header />

<section>

<div className="py-8 px-6 mx-auto max-w-screen-xl text-center lg:py-16 lg:px-12 md:px-10">

<h1 className="mt-4 lg:mt-8 mb-4 text-4xl font-extrabold tracking-tight leading-none text-black md:text-5xl lg:text-6xl">

Build Your Resume <span className="text-primary-700 max-sm:block">With AI</span>

</h1>

<p className="mb-8 text-lg font-normal text-gray-700 lg:text-xl sm:px-16 xl:px-48">

Effortlessly Craft a Professional Resume with Our AI-Powered Builder

</p>

<div className="flex flex-col space-y-4 sm:flex-row sm:justify-center sm:space-y-0 sm:space-x-4">

<Link

href={`${!user?.isSignedIn ? "/sign-up" : "/dashboard"}`}

className="relative flex h-11 w-full items-center justify-center px-6 before:absolute before:inset-0 before:rounded-full before:bg-primary-700 before:transition before:duration-300 hover:before:scale-105 active:duration-75 active:before:scale-95 sm:w-max"

>

<span className="relative text-base font-semibold text-white">

Get Started

</span>

</Link>

<Link

href="#learn-more"

className="relative flex h-11 w-full items-center justify-center px-6 before:absolute before:inset-0 before:rounded-full before:border before:border-transparent before:bg-slate-200 before:bg-gradient-to-b before:transition before:duration-300 hover:before:scale-105 active:duration-75 active:before:scale-95 sm:w-max"

>

<span className="relative text-base font-semibold text-primary">

Learn more

</span>

</Link>

</div>

</div>

</section>

<section className="py-8 px-6 mx-auto max-w-screen-xl text-center lg:py-8 lg:px-12 md:px-10">

<h2 className="font-bold text-3xl" id="learn-more">

How it Works?

</h2>

<h2 className="text-md text-gray-500">

Generate resume in just 3 steps

</h2>

<div className="mt-8 grid grid-cols-1 gap-8 text-center md:text-start md:grid-cols-2 lg:grid-cols-3 md:px-24">

<div className="flex flex-col cursor-pointer p-8 border border-gray-100 rounded-3xl bg-white shadow-xl max-md:shadow-md shadow-gray-600/10 hover:shadow-gray-600/15 transition-shadow duration-300 items-center md:items-start justify-center md:justify-start">

<AtomIcon className="h-8 w-8" />

<h2 className="mt-4 text-xl font-bold text-black">

Create Your Template

</h2>

<p className="mt-1 text-sm text-gray-600 md:text-justify">

Start by selecting the color scheme for your resume template. Our

single, professionally designed template ensures a clean and

consistent look for all users.

</p>

</div>

<div className="flex flex-col cursor-pointer p-8 border border-gray-100 rounded-3xl bg-white shadow-xl max-md:shadow-md shadow-gray-600/10 hover:shadow-gray-600/15 transition-shadow duration-300 items-center md:items-start justify-center md:justify-start">

<Edit className="h-8 w-8" />

<h2 className="mt-4 text-xl font-bold text-black">

Update Your Information

</h2>

<p className="mt-1 text-sm text-gray-600 md:text-justify">

Enter your personal details, work experience, education, and

skills into the provided form. Our AI assists you in filling out

each section accurately and effectively.

</p>

</div>

<div className="flex flex-col cursor-pointer p-8 border border-gray-100 rounded-3xl bg-white

shadow-xl max-md:shadow-md shadow-gray-600/10 hover:shadow-gray-600/15 transition-shadow duration-300 items-center md:items-start justify-center md:justify-start">

<Share2 className="h-8 w-8" />

<h2 className="mt-4 text-xl font-bold text-black">

Share Your Resume

</h2>

<p className="mt-1 text-sm text-gray-600 md:text-justify">

After completing your resume, save it securely and generate a

shareable link. Easily update your information anytime and share

the link with potential employers or download it in a preferred

format.

</p>

</div>

</div>

<div className="mt-20 text-center">

<Link

href="#get-started"

className="inline-block rounded-full bg-primary-700 px-12 py-3 text-sm font-medium text-white transition hover:bg-primary-800 focus:outline-none focus:ring focus:ring-primary-400"

>

<div className="flex items-center justify-center">

<ArrowBigUp className="h-6 w-6 mr-2" />

Get Started Today

</div>

</Link>

</div>

</section>

<footer className="backdrop-blur-md w-full">

<div className="w-full mx-auto text-center max-w-screen-xl p-4 flex max-md:flex-col md:items-center md:justify-between">

<span className="text-sm text-gray-500 sm:text-center">

© KLH{" "}

<span className="hover:text-primary-500 hover:cursor-pointer">

ResumeAI™

</span>

. All Rights Reserved.

</span>

<Link href="https://github.com/Mahmud0808" className="me-4 md:me-6">

<span className="hover:text-primary-500 mt-3 text-sm font-medium text-gray-500 sm:mt-0">

</span>

</Link>

</div>

</footer>

</div>

);

};

export default page;

**OUTPUT:**













