

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;  
  
@font-face {  
    font-family: "Manrope";  
    src: url("/fonts/Manrope-VariableFont_wght.ttf") format("truetype");  
    font-weight: 200 800;  
    font-style: normal;  
    font-display: swap;  
}  
  
body {  
    font-family: "Manrope", sans-serif;  
}  
  
@layer utilities {  
    .text-balance {  
        text-wrap: balance;  
    }  
}  
  
@layer base {  
    :root {  
        --background: 0 0% 100%;  
        --foreground: 0 0% 3.9%;  
        --card: 0 0% 100%;  
        --card-foreground: 0 0% 3.9%;  
        --popover: 0 0% 100%;  
        --popover-foreground: 0 0% 3.9%;  
        --primary: 0 0% 9%;  
        --primary-foreground: 0 0% 98%;  
        --secondary: 0 0% 96.1%;  
        --secondary-foreground: 0 0% 9%;  
        --muted: 0 0% 96.1%;  
        --muted-foreground: 0 0% 45.1%;  
        --accent: 0 0% 96.1%;  
        --accent-foreground: 0 0% 9%;  
        --destructive: 0 84.2% 60.2%;  
        --destructive-foreground: 0 0% 98%;  
        --border: 0 0% 89.8%;  
        --input: 0 0% 89.8%;  
        --ring: 0 0% 3.9%;  
        --chart-1: 12 76% 61%;  
    }  
}
```

```
--chart-2: 173 58% 39%;  
--chart-3: 197 37% 24%;  
--chart-4: 43 74% 66%;  
--chart-5: 27 87% 67%;  
--radius: 0.5rem;  
--sidebar-background: 0 0% 98%;  
--sidebar-foreground: 240 5.3% 26.1%;  
--sidebar-primary: 240 5.9% 10%;  
--sidebar-primary-foreground: 0 0% 98%;  
--sidebar-accent: 240 4.8% 95.9%;  
--sidebar-accent-foreground: 240 5.9% 10%;  
--sidebar-border: 220 13% 91%;  
--sidebar-ring: 217.2 91.2% 59.8%;  
}  
.dark {  
  --background: 0 0% 3.9%;  
  --foreground: 0 0% 98%;  
  --card: 0 0% 3.9%;  
  --card-foreground: 0 0% 98%;  
  --popover: 0 0% 3.9%;  
  --popover-foreground: 0 0% 98%;  
  --primary: 0 0% 98%;  
  --primary-foreground: 0 0% 9%;  
  --secondary: 0 0% 14.9%;  
  --secondary-foreground: 0 0% 98%;  
  --muted: 0 0% 14.9%;  
  --muted-foreground: 0 0% 63.9%;  
  --accent: 0 0% 14.9%;  
  --accent-foreground: 0 0% 98%;  
  --destructive: 0 62.8% 30.6%;  
  --destructive-foreground: 0 0% 98%;  
  --border: 0 0% 14.9%;  
  --input: 0 0% 14.9%;  
  --ring: 0 0% 83.1%;  
  --chart-1: 220 70% 50%;  
  --chart-2: 160 60% 45%;  
  --chart-3: 30 80% 55%;  
  --chart-4: 280 65% 60%;  
  --chart-5: 340 75% 55%;  
  --sidebar-background: 240 5.9% 10%;  
  --sidebar-foreground: 240 4.8% 95.9%;  
  --sidebar-primary: 224.3 76.3% 48%;  
  --sidebar-primary-foreground: 0 0% 100%;  
  --sidebar-accent: 240 3.7% 15.9%;
```



```

        { id: "chat", icon: MessageCircle, label: "Chat History" },
        { id: "journal", icon: BookOpen, label: "Journal" },
        { id: "tasks", icon: CheckSquare, label: "Task List" },
    ]
}

return (
  <AnimatePresence>
  {isOpen && (
    <motion.div
      initial={{ opacity: 0, y: -20, height: 0 }}
      animate={{ opacity: 1, y: 0, height: "auto" }}
      exit={{ opacity: 0, y: -20, height: 0 }}
      transition={{ duration: 0.3, ease: [0.16, 1, 0.3, 1] }}
      className="absolute top-full left-0 right-0 mt-2 overflow-hidden"
    >
      <div className="relative bg-brand-light/[0.08] backdrop-blur-3xl border border-brand-light/15 rounded-2xl shadow-2xl">
        <div className="absolute inset-0 rounded-2xl border border-brand-primary/10 pointer-events-none" />

        <div className="flex">
          {/* Content Area */}
          <div className="flex-1 p-4">
            {activeTab === "chat" && (
              <div className="space-y-3 max-h-64 overflow-y-auto">
                {messages.length === 0 ? (
                  <div className="text-center text-brand-light/60 py-8">
                    <MessageCircle className="w-8 h-8 mx-auto mb-2 opacity-50" />
                    <p className="text-sm">No messages yet. Start a conversation!</p>
                  </div>
                ) : (
                  messages.map((message) => (
                    <motion.div
                      key={message.id}
                      initial={{ opacity: 0, y: 10 }}
                      animate={{ opacity: 1, y: 0 }}
                      className={`p-3 rounded-lg ${message.type === "user"
                        ? "bg-brand-primary/10 border border-brand-primary/20 ml-8"
                        : "bg-brand-light/[0.04] border border-brand-light/[0.06] mr-8"
                      }`}
                  >
                    <div className="text-brand-light text-sm font-medium mb-1">
                      {message.type === "user" ? "You" : "Assistant"}
                    </div>
                  </motion.div>
                )
              )
            )}
          </div>
        </div>
      </div>
    </motion.div>
  ) : null
)

```

```

        </div>
        <div className="text-brand-light/80 text-sm">{message.content}</div>
        <div className="text-brand-light/40 text-xs mt-2">
            {message.timestamp.toLocaleTimeString()}
        </div>
    </motion.div>
)
}
</div>
}

{activeTab === "journal" && (
    <div className="text-center text-brand-light/60 py-8">
        <BookOpen className="w-8 h-8 mx-auto mb-2 opacity-50" />
        <p className="text-sm">Journal feature coming soon</p>
    </div>
)

{activeTab === "tasks" && (
    <div className="text-center text-brand-light/60 py-8">
        <CheckSquare className="w-8 h-8 mx-auto mb-2 opacity-50" />
        <p className="text-sm">Task list feature coming soon</p>
    </div>
)
}
</div>

/* Sidebar */
<div className="w-16 bg-brand-primary/10 backdrop-blur-sm border-l border-brand-light/10 rounded-r-2xl p-2">
    <div className="space-y-2">
        {tabs.map((tab) => (
            <motion.button
                key={tab.id}
                onClick={() => setActiveTab(tab.id)}
                whileHover={{ scale: 1.05 }}
                whileTap={{ scale: 0.95 }}
                className={`w-12 h-12 rounded-xl flex items-center justify-center transition-all duration-200 ${activeTab === tab.id ? "bg-brand-primary/20 border border-brand-primary/30" : "bg-brand-light/[0.06] hover:bg-brand-primary/10 border border-brand-light/10"}`}
            >
                <tab.icon

```

```

        className={`${w-5 h-5 ${activeTab === tab.id ? "text-brand-accent" : "text-brand-light/70"}`}
        strokeWidth={1.5}
      />
    </motion.button>
  )})
</div>
</div>
</div>
</div>
</motion.div>
)}
</AnimatePresence>
)
}

```

"use client"

```

import { motion, AnimatePresence } from "framer-motion"
import { Search, Globe, FileText, Clipboard, Grid3X3 } from "lucide-react"
import { useState } from "react"

```

```

interface ContextMenuProps {
  isOpen: boolean
  onClose: () => void
}

```

```

export function ContextMenu({ isOpen, onClose }: ContextMenuProps) {
  const [activeFilter, setActiveFilter] = useState("all")
  const [searchQuery, setSearchQuery] = useState("")

```

```

  const filters = [
    { id: "web", icon: Globe, label: "Web" },
    { id: "files", icon: FileText, label: "Files" },
    { id: "clipboard", icon: Clipboard, label: "Clipboard" },
    { id: "all", icon: Grid3X3, label: "All" },
  ]

```

```

  const contextItems = {
    web: [
      { title: "Recent Search", content: "Machine learning tutorials" },
      { title: "Bookmarked Page", content: "React documentation" },
    ],
    files: [

```

```

        { title: "Document.pdf", content: "Project specifications" },
        { title: "Notes.txt", content: "Meeting notes from yesterday" },
    ],
    clipboard: [
        { title: "Copied Text", content: "const handleSubmit = () => {" },
        { title: "URL", content: "https://example.com/api/docs" },
    ],
    all: [
        { title: "Recent Search", content: "Machine learning tutorials" },
        { title: "Document.pdf", content: "Project specifications" },
        { title: "Copied Text", content: "const handleSubmit = () => {" },
    ],
}
}

return (
<AnimatePresence>
{isOpen && (
<motion.div
    initial={{ opacity: 0, y: 20, height: 0 }}
    animate={{ opacity: 1, y: 0, height: "auto" }}
    exit={{ opacity: 0, y: 20, height: 0 }}
    transition={{ duration: 0.3, ease: [0.16, 1, 0.3, 1] }}
    className="absolute bottom-full left-0 right-0 mb-2 overflow-hidden"
>
    <div className="relative bg-brand-light/[0.08] backdrop-blur-3xl border
border-brand-light/15 rounded-2xl shadow-2xl">
        <div className="absolute inset-0 rounded-2xl border border-brand-primary/10
pointer-events-none" />

        <div className="flex">
            {/* Filter Sidebar */}
            <div className="w-16 bg-brand-primary/10 backdrop-blur-sm border-
border-brand-light/10 rounded-l-2xl p-2">
                <div className="space-y-2">
                    {filters.map((filter) => (
                        <motion.button
                            key={filter.id}
                            onClick={() => setActiveFilter(filter.id)}
                            whileHover={{ scale: 1.05 }}
                            whileTap={{ scale: 0.95 }}
                            className={`${`w-12 h-12 rounded-xl flex items-center justify-center transition-all
duration-200 ${activeFilter === filter.id ? "bg-brand-primary/20 border border-brand-primary/30" : ""}`}`}
                    )}
                </div>
            </div>
        </div>
    </div>

```

```

        : "bg-brand-light/[0.06] hover:bg-brand-primary/10 border border-brand-light/10"
    )}
>
<filter.icon
  className={`${w-5 h-5 ${
    activeFilter === filter.id ? "text-brand-accent" : "text-brand-light/70"
  }}`}
  strokeWidth={1.5}
/>
</motion.button>
)})}
</div>
</div>

/* Content Area */
<div className="flex-1 p-4">
/* Search Bar */
<div className="relative mb-4">
<div className="absolute inset-0 bg-brand-light/[0.06] backdrop-blur-sm rounded-xl border border-brand-light/[0.08]" />
<div className="relative flex items-center gap-3 p-3">
<Search className="w-4 h-4 text-brand-light/60" strokeWidth={1.5} />
<input
  type="text"
  placeholder="Filter context..."
  value={searchQuery}
  onChange={(e) => setSearchQuery(e.target.value)}
  className="flex-1 bg-transparent text-brand-light placeholder-brand-light/40 outline-none font-normal text-sm"
/>
</div>
</div>

/* Context Items */
<div className="space-y-2 max-h-48 overflow-y-auto">
{contextItems[activeFilter] as keyof typeof contextItems}
.filter(
  (item) =>
    item.title.toLowerCase().includes(searchQuery.toLowerCase()) ||
    item.content.toLowerCase().includes(searchQuery.toLowerCase()),
)
.map((item, index) => (
<motion.div
  key={index}

```

```

    initial={{ opacity: 0, y: 10 }}
    animate={{ opacity: 1, y: 0 }}
    transition={{ delay: index * 0.05 }}
    className="p-3 bg-brand-light/[0.04] hover:bg-brand-primary/[0.08] rounded-lg
border border-brand-light/[0.06] cursor-pointer transition-all duration-200"
  >
  <div className="text-brand-light font-medium text-sm mb-1">{item.title}</div>
  <div className="text-brand-light/60 text-xs truncate">{item.content}</div>
</motion.div>
))
</div>
</div>
</div>
</div>
</div>
</motion.div>
)}
</AnimatePresence>
)
}

```

"use client"

```

import type React from "react"

import { useState } from "react"
import { motion, AnimatePresence } from "framer-motion"
import { MessageCircle, Mic, Camera, History, Menu, Send } from "lucide-react"
import { ContextMenu } from "./components/context-menu"
import { ChatView } from "./components/chat-view"

```

```

interface Message {
  id: string
  type: "user" | "bot"
  content: string
  timestamp: Date
}

```

```

export default function Component() {
  const [isContextMenuOpen, setIsContextMenuOpen] = useState(false)
  const [isChatViewOpen, setIsChatViewOpen] = useState(false)
  const [inputValue, setInputValue] = useState("")
  const [messages, setMessages] = useState<Message[]>([])
  const [isListening, setIsListening] = useState(false)
}

```

```

const handleSendMessage = () => {
  if (!inputValue.trim()) return

  const userMessage: Message = {
    id: Date.now().toString(),
    type: "user",
    content: inputValue,
    timestamp: new Date(),
  }

  const botMessage: Message = {
    id: (Date.now() + 1).toString(),
    type: "bot",
    content: "I'm processing your request. This is a demo response.",
    timestamp: new Date(),
  }

  setMessages([...prev, userMessage, botMessage])
  setInputValue("")
  setIsChatViewOpen(true)
}

const handleKeyPress = (e: React.KeyboardEvent) => {
  if (e.key === "Enter" && !e.shiftKey) {
    e.preventDefault()
    handleSendMessage()
  }
}

return (
  <div className="min-h-screen bg-brand-dark relative overflow-hidden font-manrope">
    /* Background Effects */
    <div className="fixed inset-0 opacity-10">
      <div className="absolute inset-0
        bg-[radial-gradient(circle_at_25%_25%,#123524,transparent_50%)]" />
      <div className="absolute inset-0
        bg-[radial-gradient(circle_at_75%_75%,#D64933,transparent_70%)]" />
      <div className="absolute inset-0
        bg-[linear-gradient(45deg,transparent_48%,rgba(18,53,36,0.1)_50%,transparent_52%)]" />
    </div>

    <div className="fixed inset-0 pointer-events-none">
      <div className="absolute top-1/4 left-1/4 w-96 h-96 bg-brand-primary/5 rounded-full
        blur-3xl" />
    
```

```

<div className="absolute bottom-1/4 right-1/4 w-96 h-96 bg-brand-accent/3 rounded-full blur-3xl" />
</div>

<div className="relative z-10 flex items-center justify-center min-h-screen p-6">
<motion.div
  initial={{ opacity: 0, scale: 0.95, y: 20 }}
  animate={{ opacity: 1, scale: 1, y: 0 }}
  transition={{ duration: 0.8, ease: [0.16, 1, 0.3, 1] }}
  className="relative max-w-lg w-full"
>
  {/* Outer Glow */}
  <div className="absolute -inset-1 bg-gradient-to-r from-brand-primary/20 via-brand-light/10 to-brand-accent/20 rounded-3xl blur-xl opacity-60" />
  <div className="absolute -inset-2 bg-brand-primary/5 rounded-3xl blur-2xl" />

  {/* Main Container */}
  <div className="relative">
    {/* Context Menu */}
    <ContextMenu isOpen={isContextMenuOpen} onClose={() => setIsContextMenuOpen(false)} />

    {/* Main Input Panel */}
    <div className="relative bg-brand-light/[0.08] backdrop-blur-3xl border border-brand-light/15 rounded-3xl shadow-2xl overflow-hidden">
      <div className="absolute inset-0 rounded-3xl border border-brand-primary/10 pointer-events-none" />

      {/* Rainbow border animation when listening */}
      <AnimatePresence>
        {isListening && (
          <motion.div
            initial={{ opacity: 0 }}
            animate={{ opacity: 1 }}
            exit={{ opacity: 0 }}
            className="absolute inset-0 rounded-3xl border-2 border-transparent bg-gradient-to-r from-red-500 via-yellow-500 via-green-500 via-blue-500 via-indigo-500 to-purple-500 animate-pulse"
            style={{
              background:
                "linear-gradient(45deg, #ff0000, #ff7f00, #ffff00, #00ff00, #0000ff, #4b0082, #9400d3)",
              backgroundSize: "400% 400%",
              animation: "rainbow 2s linear infinite",
            }
          </motion.div>
        )}
      </AnimatePresence>
    </div>
  </div>
</motion.div>
</div>

```

```

        }
    />
})
</AnimatePresence>

{/* Header */}
<div className="flex items-center justify-between p-6 border-b border-brand-light/[0.08]">
    <div className="flex items-center gap-3">
        <div className="relative">
            <div className="absolute inset-0 bg-brand-primary/20 rounded-xl blur-sm" />
            <div className="relative w-10 h-10 bg-brand-primary/20 backdrop-blur-sm rounded-xl flex items-center justify-center border border-brand-primary/30">
                <MessageCircle className="w-5 h-5 text-brand-light" strokeWidth={1.5} />
            </div>
        </div>
        <h1 className="text-xl font-medium text-brand-light tracking-wide">SENTER</h1>
    </div>
</div>

{/* Enhanced Input Bar */}
<div className="p-6">
    <div className="relative">
        <div className="absolute inset-0 bg-brand-light/[0.06] backdrop-blur-sm rounded-2xl border border-brand-light/[0.08]" />
        <div className="relative flex items-center gap-3 p-4">
            {/* Context Menu Button */}
            <motion.button
                onClick={() => setIsContextMenuOpen(!isContextMenuOpen)}
                whileHover={{ scale: 1.05 }}
                whileTap={{ scale: 0.95 }}
                className={`w-8 h-8 rounded-lg flex items-center justify-center transition-all duration-200 ${isContextMenuOpen ? "bg-brand-primary/20 border border-brand-primary/30" : "bg-brand-light/[0.08] hover:bg-brand-primary/10 border border-brand-light/10"}`}
            >
                <Menu
                    className={`${`w-4 h-4 ${isContextMenuOpen ? "text-brand-accent" : "text-brand-light/70"}`}`}
                    strokeWidth={1.5}
                />
            </motion.button>
        </div>
    </div>
</div>

```

```

/* Input Field */
<textarea
  value={inputValue}
  onChange={(e) => setInputValue(e.target.value)}
  onKeyPress={handleKeyPress}
  placeholder="Ask me anything..."
  className="flex-1 bg-transparent text-brand-light placeholder-brand-light/40
outline-none font-normal text-sm resize-none min-h-[20px] max-h-24"
  rows={1}
/>

/* Action Buttons */
<div className="flex items-center gap-2">
  /* Microphone Button */
  <motion.button
    onMouseDown={() => setIsListening(true)}
    onMouseUp={() => setIsListening(false)}
    onMouseLeave={() => setIsListening(false)}
    whileHover={{ scale: 1.05 }}
    whileTap={{ scale: 0.95 }}
    className={`${w-8 h-8 rounded-lg flex items-center justify-center transition-all
duration-200 ${`isListening ? "text-brand-accent" : "text-brand-light/70"}`}
    isListening
      ? "bg-brand-accent/20 border border-brand-accent/30"
      : "bg-brand-light/[0.08] hover:bg-brand-primary/10 border
border-brand-light/10"
  }>
    <Mic
      className={`${w-4 h-4 ${`isListening ? "text-brand-accent" :
"text-brand-light/70"}`}`}
      strokeWidth={1.5}
    />
  </motion.button>

  /* Camera Button */
  <motion.button
    whileHover={{ scale: 1.05 }}
    whileTap={{ scale: 0.95, backgroundColor: "rgba(214, 73, 51, 0.2)" }}
    className="w-8 h-8 rounded-lg flex items-center justify-center
bg-brand-light/[0.08] hover:bg-brand-primary/10 border border-brand-light/10 transition-all
duration-200"
  >

```

```

        <Camera className="w-4 h-4 text-brand-light/70" strokeWidth={1.5} />
    </motion.button>

    {/* Send Button */}
    <motion.button
        onClick={handleSendMessage}
        whileHover={{ scale: 1.05 }}
        whileTap={{ scale: 0.95 }}
        disabled={!inputValue.trim()}
        className="w-8 h-8 rounded-lg flex items-center justify-center
        bg-brand-accent/90 hover:bg-brand-accent disabled:opacity-50 disabled:cursor-not-allowed
        border border-brand-accent/20 transition-all duration-200"
    >
        <Send className="w-4 h-4 text-brand-light" strokeWidth={1.5} />
    </motion.button>

    {/* Chat History Button */}
    <motion.button
        onClick={() => setIsChatViewOpen(!isChatViewOpen)}
        whileHover={{ scale: 1.05 }}
        whileTap={{ scale: 0.95 }}
        className={`${`w-8 h-8 rounded-lg flex items-center justify-center transition-all
        duration-200 ${`isChatViewOpen
            ? "bg-brand-primary/20 border border-brand-primary/30"
            : "bg-brand-light/[0.08] hover:bg-brand-primary/10 border
        border-brand-light/10"
        }}`}
    >
        <History
            className={`${`w-4 h-4 ${`isChatViewOpen ? "text-brand-accent" :
        "text-brand-light/70"`${`}}`}
            strokeWidth={1.5}
        />
    </motion.button>
</div>
</div>
</div>
</div>
</div>

    {/* Chat View */}
    <ChatView isOpen={isChatViewOpen} messages={messages} onClose={() =>
        setIsChatViewOpen(false)} />

```

```
</div>
</motion.div>
</div>

<style jsx>{
  @keyframes rainbow {
    0% { background-position: 0% 50%; }
    50% { background-position: 100% 50%; }
    100% { background-position: 0% 50%; }
  }
}</style>
</div>
)

}

import type { Config } from "tailwindcss"

// all in fixtures is set to tailwind v3 as interims solutions

const config: Config = {
  darkMode: ["class"],
  content: [
    "./pages/**/*.{js,ts,jsx,tsx,mdx}",
    "./components/**/*.{js,ts,jsx,tsx,mdx}",
    "./app/**/*.{js,ts,jsx,tsx,mdx}",
    "*.{js,ts,jsx,tsx,mdx}",
  ],
  theme: {
    extend: {
      fontFamily: {
        manrope: ["Manrope", "sans-serif"],
      },
      colors: {
        background: "hsl(var(--background))",
        foreground: "hsl(var(--foreground))",
        card: {
          DEFAULT: "hsl(var(--card))",
          foreground: "hsl(var(--card-foreground))",
        },
        popover: {
          DEFAULT: "hsl(var(--popover))",
          foreground: "hsl(var(--popover-foreground))",
        },
        primary: {
```

```
    DEFAULT: "hsl(var(--primary))",
    foreground: "hsl(var(--primary-foreground))",
  },
  secondary: {
    DEFAULT: "hsl(var(--secondary))",
    foreground: "hsl(var(--secondary-foreground))",
  },
  muted: {
    DEFAULT: "hsl(var(--muted))",
    foreground: "hsl(var(--muted-foreground))",
  },
  accent: {
    DEFAULT: "hsl(var(--accent))",
    foreground: "hsl(var(--accent-foreground))",
  },
  destructive: {
    DEFAULT: "hsl(var(--destructive))",
    foreground: "hsl(var(--destructive-foreground))",
  },
  border: "hsl(var(--border))",
  input: "hsl(var(--input))",
  ring: "hsl(var(--ring))",
  chart: {
    "1": "hsl(var(--chart-1))",
    "2": "hsl(var(--chart-2))",
    "3": "hsl(var(--chart-3))",
    "4": "hsl(var(--chart-4))",
    "5": "hsl(var(--chart-5))",
  },
  sidebar: {
    DEFAULT: "hsl(var(--sidebar-background))",
    foreground: "hsl(var(--sidebar-foreground))",
    primary: "hsl(var(--sidebar-primary))",
    "primary-foreground": "hsl(var(--sidebar-primary-foreground))",
    accent: "hsl(var(--sidebar-accent))",
    "accent-foreground": "hsl(var(--sidebar-accent-foreground))",
    border: "hsl(var(--sidebar-border))",
    ring: "hsl(var(--sidebar-ring))",
  },
  brand: {
    primary: "#123524",
    dark: "#081710",
    light: "#FAFAFA",
    accent: "#D64933",
```

```
        },
      },
      borderRadius: {
        lg: "var(--radius)",
        md: "calc(var(--radius) - 2px)",
        sm: "calc(var(--radius) - 4px)",
      },
      keyframes: {
        "accordion-down": {
          from: {
            height: "0",
          },
          to: {
            height: "var(--radix-accordion-content-height)",
          },
        },
        "accordion-up": {
          from: {
            height: "var(--radix-accordion-content-height)",
          },
          to: {
            height: "0",
          },
        },
      },
      animation: {
        "accordion-down": "accordion-down 0.2s ease-out",
        "accordion-up": "accordion-up 0.2s ease-out",
      },
      backgroundImage: {
        "gradient-radial": "radial-gradient(var(--tw-gradient-stops))",
        "gradient-conic": "conic-gradient(from 180deg at 50% 50%, var(--tw-gradient-stops))",
      },
    },
    plugins: [],
  }
export default config
```