Build this as my initial prototype

Copy-paste this component to /components/ui folder:
```tsx
pricing.tsx
"use client";

import { buttonVariants } from "@/components/ui/button";
import { Label } from "@/components/ui/label";
import { Switch } from "@/components/ui/switch";
import { useMediaQuery } from "@/hooks/use-media-query";
import { cn } from "@/lib/utils";
import { motion } from "framer-motion";
import { Check, Star } from "lucide-react";
import Link from "next/link";
import { useState, useRef } from "react";
import confetti from "canvas-confetti";
import NumberFlow from "@number-flow/react";

interface PricingPlan {
  name: string;
  price: string;
  yearlyPrice: string;
  period: string;
  features: string[];
  description: string;
  buttonText: string;
  href: string;
  isPopular: boolean;
}

interface PricingProps {
  plans: PricingPlan[];
  title?: string;
  description?: string;
}

export function Pricing({
  plans,
  title = "Simple, Transparent Pricing",
  description = "Choose the plan that works for you
All plans include access to our platform, lead generation tools, and dedicated support.",
}: PricingProps) {
  const [isMonthly, setIsMonthly] = useState(true);
```

```
const isDesktop = useMediaQuery("(min-width: 768px)");
const switchRef = useRef<HTMLButtonElement>(null);

const handleToggle = (checked: boolean) => {
  setIsMonthly(!checked);
  if (checked && switchRef.current) {
    const rect = switchRef.current.getBoundingClientRect();
    const x = rect.left + rect.width / 2;
    const y = rect.top + rect.height / 2;

    confetti({
      particleCount: 50,
      spread: 60,
      origin: {
        x: x / window.innerWidth,
        y: y / window.innerHeight,
      },
      colors: [
        "hsl(var(--primary))",
        "hsl(var(--accent))",
        "hsl(var(--secondary))",
        "hsl(var(--muted))",
      ],
      ticks: 200,
      gravity: 1.2,
      decay: 0.94,
      startVelocity: 30,
      shapes: ["circle"],
    });
  }
};

return (
  <div className="container py-20">
    <div className="text-center space-y-4 mb-12">
      <h2 className="text-4xl font-bold tracking-tight sm:text-5xl">
        {title}
      </h2>
      <p className="text-muted-foreground text-lg whitespace-pre-line">
        {description}
      </p>
    </div>

    <div className="flex justify-center mb-10">
```

```
        <label className="relative inline-flex items-center cursor-pointer">
          <Label>
            <Switch
              ref={switchRef as any}
              checked={!isMonthly}
              onCheckedChange={handleToggle}
              className="relative"
            />
          </Label>
        </label>
        <span className="ml-2 font-semibold">
          Annual billing <span className="text-primary">(Save 20%)</span>
        </span>
      </div>

      <div className="grid grid-cols-1 md:grid-cols-3 sm:2 gap-4">
        {plans.map((plan, index) => (
          <motion.div
            key={index}
            initial={{ y: 50, opacity: 1 }}
            whileInView={
              isDesktop
                ? {
                    y: plan.isPopular ? -20 : 0,
                    opacity: 1,
                    x: index === 2 ? -30 : index === 0 ? 30 : 0,
                    scale: index === 0 || index === 2 ? 0.94 : 1.0,
                  }
                : {}
            }
            viewport={{ once: true }}
            transition={{
              duration: 1.6,
              type: "spring",
              stiffness: 100,
              damping: 30,
              delay: 0.4,
              opacity: { duration: 0.5 },
            }}
            className={cn(
              `rounded-2xl border-[1px] p-6 bg-background text-center lg:flex lg:flex-col
lg:justify-center relative`,
              plan.isPopular ? "border-primary border-2" : "border-border",
              "flex flex-col",
```

```
            !plan.isPopular && "mt-5",
            index === 0 || index === 2
              ? "z-0 transform translate-x-0 translate-y-0 -translate-z-[50px] rotate-y-[10deg]"
              : "z-10",
            index === 0 && "origin-right",
            index === 2 && "origin-left"
          )}
        >
          {plan.isPopular && (
            <div classNamе="absolute top-0 right-0 bg-primary py-0.5 px-2 rounded-bl-xl
rounded-tr-xl flex items-center">
              <Star classNamе="text-primary-foreground h-4 w-4 fill-current" />
              <span classNamе="text-primary-foreground ml-1 font-sans font-semibold">
                Popular
              </span>
            </div>
          )}
          <div classNamе="flex-1 flex flex-col">
            <p classNamе="text-base font-semibold text-muted-foreground">
              {plan.name}
            </p>
            <div classNamе="mt-6 flex items-center justify-center gap-x-2">
              <span classNamе="text-5xl font-bold tracking-tight text-foreground">
                <NumberFlow
                  value={
                    isMonthly ? Number(plan.price) : Number(plan.yearlyPrice)
                  }
                  format={{
                    style: "currency",
                    currency: "USD",
                    minimumFractionDigits: 0,
                    maximumFractionDigits: 0,
                  }}
                  formatter={(value) => `$${value}`}
                  transformTiming={{
                    duration: 500,
                    easing: "ease-out",
                  }}
                  willChange
                  classNamе="font-variant-numeric: tabular-nums"
                />
              </span>
              {plan.period !== "Next 3 months" && (
```

```
              <span className="text-sm font-semibold leading-6 tracking-wide
text-muted-foreground">
                / {plan.period}
              </span>
            )}
          </div>

          <p className="text-xs leading-5 text-muted-foreground">
            {isMonthly ? "billed monthly" : "billed annually"}
          </p>

          <ul className="mt-5 gap-2 flex flex-col">
            {plan.features.map((feature, idx) => (
              <li key={idx} className="flex items-start gap-2">
                <Check className="h-4 w-4 text-primary mt-1 flex-shrink-0" />
                <span className="text-left">{feature}</span>
              </li>
            ))}
          </ul>

          <hr className="w-full my-4" />

          <Link
            href={plan.href}
            className={cn(
              buttonVariants({
                variant: "outline",
              }),
              "group relative w-full gap-2 overflow-hidden text-lg font-semibold tracking-tighter",
              "transform-gpu ring-offset-current transition-all duration-300 ease-out hover:ring-2
hover:ring-primary hover:ring-offset-1 hover:bg-primary hover:text-primary-foreground",
              plan.isPopular
                ? "bg-primary text-primary-foreground"
                : "bg-background text-foreground"
            )}
          >
            {plan.buttonText}
          </Link>
          <p className="mt-6 text-xs leading-5 text-muted-foreground">
            {plan.description}
          </p>
        </div>
      </motion.div>
    ))}
```

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

demo.tsx

```
"use client";

import { Pricing } from "@/components/blocks/pricing";

const demoPlans = [
  {
    name: "STARTER",
    price: "50",
    yearlyPrice: "40",
    period: "per month",
    features: [
      "Up to 10 projects",
      "Basic analytics",
      "48-hour support response time",
      "Limited API access",
      "Community support",
    ],
    description: "Perfect for individuals and small projects",
    buttonText: "Start Free Trial",
    href: "/sign-up",
    isPopular: false,
  },
  {
    name: "PROFESSIONAL",
    price: "99",
    yearlyPrice: "79",
    period: "per month",
    features: [
      "Unlimited projects",
      "Advanced analytics",
      "24-hour support response time",
      "Full API access",
      "Priority support",
      "Team collaboration",
      "Custom integrations",
    ],
    description: "Ideal for growing teams and businesses",
```

```
    buttonText: "Get Started",
    href: "/sign-up",
    isPopular: true,
  },
  {
    name: "ENTERPRISE",
    price: "299",
    yearlyPrice: "239",
    period: "per month",
    features: [
      "Everything in Professional",
      "Custom solutions",
      "Dedicated account manager",
      "1-hour support response time",
      "SSO Authentication",
      "Advanced security",
      "Custom contracts",
      "SLA agreement",
    ],
    description: "For large organizations with specific needs",
    buttonText: "Contact Sales",
    href: "/contact",
    isPopular: false,
  },
];

function PricingBasic() {
  return (
    <div className="h-[800px] overflow-y-auto rounded-lg">
      <Pricing
        plans={demoPlans}
        title="Simple, Transparent Pricing"
        description="Choose the plan that works for you
All plans include access to our platform, lead generation tools, and dedicated support."
      />
    </div>
  );
}

export { PricingBasic };

```

Copy-paste these files for dependencies:

```tsx
shadcn/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-background transition-colors focus-visible:outline-none focus-visible:ring-2
focus-visible:ring-ring focus-visible:ring-offset-2 disabled:pointer-events-none
disabled:opacity-50",
  {
    variants: {
      variant: {
        default: "bg-primary text-primary-foreground hover:bg-primary/90",
        destructive:
          "bg-destructive text-destructive-foreground hover:bg-destructive/90",
        outline:
          "border border-input bg-background hover:bg-accent hover:text-accent-foreground",
        secondary:
          "bg-secondary text-secondary-foreground hover:bg-secondary/80",
        ghost: "hover:bg-accent hover:text-accent-foreground",
        link: "text-primary underline-offset-4 hover:underline",
      },
      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 }

```
```tsx
shadcn/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 }

```
```tsx
shadcn/switch
"use client"

import * as React from "react"
import * as SwitchPrimitives from "@radix-ui/react-switch"

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

const Switch = React.forwardRef<
  React.ElementRef<typeof SwitchPrimitives.Root>,
  React.ComponentPropsWithoutRef<typeof SwitchPrimitives.Root>
>(({ className, ...props }, ref) => (
  <SwitchPrimitives.Root
    className={cn(
      "peer inline-flex h-6 w-11 shrink-0 cursor-pointer items-center rounded-full border-2
border-transparent transition-colors focus-visible:outline-none focus-visible:ring-2
focus-visible:ring-ring focus-visible:ring-offset-2 focus-visible:ring-offset-background
disabled:cursor-not-allowed disabled:opacity-50 data-[state=checked]:bg-primary
data-[state=unchecked]:bg-input",
      className,
    )}
    {...props}
    ref={ref}
  >
    <SwitchPrimitives.Thumb
      className={cn(
        "pointer-events-none block h-5 w-5 rounded-full bg-background shadow-lg ring-0
transition-transform data-[state=checked]:translate-x-5 data-[state=unchecked]:translate-x-0",
      )}
    />
  </SwitchPrimitives.Root>
))
Switch.displayName = SwitchPrimitives.Root.displayName

export { Switch }

```
```
```

Install these NPM dependencies:
```bash
lucide-react, framer-motion, canvas-confetti, @number-flow/react, @radix-ui/react-slot,
class-variance-authority, @radix-ui/react-label, @radix-ui/react-switch
```