Build this as my initial prototype

Copy-paste this component to /components/ui folder:

```tsx

v0-ai-chat.tsx

"use client";

import { useEffect, useRef, useCallback } from "react";

import { useState } from "react";

import { Textarea } from "@/components/ui/textarea";

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

import {

ImageIcon,

FileUp,

Figma,

MonitorIcon,

CircleUserRound,

ArrowUpIcon,

Paperclip,

PlusIcon,

} from "lucide-react";

interface UseAutoResizeTextareaProps {

minHeight: number;

maxHeight?: number;

}

function useAutoResizeTextarea({

minHeight,

maxHeight,

}: UseAutoResizeTextareaProps) {

const textareaRef = useRef<HTMLTextAreaElement>(null);

const adjustHeight = useCallback(

(reset?: boolean) => {

const textarea = textareaRef.current;

if (!textarea) return;

if (reset) {

textarea.style.height = `${minHeight}px`;

return;

}

// Temporarily shrink to get the right scrollHeight

textarea.style.height = `${minHeight}px`;

// Calculate new height

const newHeight = Math.max(

minHeight,

Math.min(

textarea.scrollHeight,

maxHeight ?? Number.POSITIVE\_INFINITY

)

);

textarea.style.height = `${newHeight}px`;

},

[minHeight, maxHeight]

);

useEffect(() => {

// Set initial height

const textarea = textareaRef.current;

if (textarea) {

textarea.style.height = `${minHeight}px`;

}

}, [minHeight]);

// Adjust height on window resize

useEffect(() => {

const handleResize = () => adjustHeight();

window.addEventListener("resize", handleResize);

return () => window.removeEventListener("resize", handleResize);

}, [adjustHeight]);

return { textareaRef, adjustHeight };

}

export function VercelV0Chat() {

const [value, setValue] = useState("");

const { textareaRef, adjustHeight } = useAutoResizeTextarea({

minHeight: 60,

maxHeight: 200,

});

const handleKeyDown = (e: React.KeyboardEvent<HTMLTextAreaElement>) => {

if (e.key === "Enter" && !e.shiftKey) {

e.preventDefault();

if (value.trim()) {

setValue("");

adjustHeight(true);

}

}

};

return (

<div className="flex flex-col items-center w-full max-w-4xl mx-auto p-4 space-y-8">

<h1 className="text-4xl font-bold text-black dark:text-white">

What can I help you ship?

</h1>

<div className="w-full">

<div className="relative bg-neutral-900 rounded-xl border border-neutral-800">

<div className="overflow-y-auto">

<Textarea

ref={textareaRef}

value={value}

onChange={(e) => {

setValue(e.target.value);

adjustHeight();

}}

onKeyDown={handleKeyDown}

placeholder="Ask v0 a question..."

className={cn(

"w-full px-4 py-3",

"resize-none",

"bg-transparent",

"border-none",

"text-white text-sm",

"focus:outline-none",

"focus-visible:ring-0 focus-visible:ring-offset-0",

"placeholder:text-neutral-500 placeholder:text-sm",

"min-h-[60px]"

)}

style={{

overflow: "hidden",

}}

/>

</div>

<div className="flex items-center justify-between p-3">

<div className="flex items-center gap-2">

<button

type="button"

className="group p-2 hover:bg-neutral-800 rounded-lg transition-colors flex items-center gap-1"

>

<Paperclip className="w-4 h-4 text-white" />

<span className="text-xs text-zinc-400 hidden group-hover:inline transition-opacity">

Attach

</span>

</button>

</div>

<div className="flex items-center gap-2">

<button

type="button"

className="px-2 py-1 rounded-lg text-sm text-zinc-400 transition-colors border border-dashed border-zinc-700 hover:border-zinc-600 hover:bg-zinc-800 flex items-center justify-between gap-1"

>

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

Project

</button>

<button

type="button"

className={cn(

"px-1.5 py-1.5 rounded-lg text-sm transition-colors border border-zinc-700 hover:border-zinc-600 hover:bg-zinc-800 flex items-center justify-between gap-1",

value.trim()

? "bg-white text-black"

: "text-zinc-400"

)}

>

<ArrowUpIcon

className={cn(

"w-4 h-4",

value.trim()

? "text-black"

: "text-zinc-400"

)}

/>

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

</button>

</div>

</div>

</div>

<div className="flex items-center justify-center gap-3 mt-4">

<ActionButton

icon={<ImageIcon className="w-4 h-4" />}

label="Clone a Screenshot"

/>

<ActionButton

icon={<Figma className="w-4 h-4" />}

label="Import from Figma"

/>

<ActionButton

icon={<FileUp className="w-4 h-4" />}

label="Upload a Project"

/>

<ActionButton

icon={<MonitorIcon className="w-4 h-4" />}

label="Landing Page"

/>

<ActionButton

icon={<CircleUserRound className="w-4 h-4" />}

label="Sign Up Form"

/>

</div>

</div>

</div>

);

}

interface ActionButtonProps {

icon: React.ReactNode;

label: string;

}

function ActionButton({ icon, label }: ActionButtonProps) {

return (

<button

type="button"

className="flex items-center gap-2 px-4 py-2 bg-neutral-900 hover:bg-neutral-800 rounded-full border border-neutral-800 text-neutral-400 hover:text-white transition-colors"

>

{icon}

<span className="text-xs">{label}</span>

</button>

);

}

demo.tsx

import { VercelV0Chat } from "@/components/ui/v0-ai-chat"

export function Demo() {

return <VercelV0Chat />

}

```

Copy-paste these files for dependencies:

```tsx

shadcn/textarea

import \* as React from "react"

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

export interface TextareaProps

extends React.TextareaHTMLAttributes<HTMLTextAreaElement> {}

const Textarea = React.forwardRef<HTMLTextAreaElement, TextareaProps>(

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

return (

<textarea

className={cn(

"flex min-h-[80px] w-full rounded-md border border-input bg-background px-3 py-2 text-sm ring-offset-background placeholder:text-muted-foreground focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50",

className

)}

ref={ref}

{...props}

/>

)

}

)

Textarea.displayName = "Textarea"

export { Textarea }

```

Install these NPM dependencies:

```bash

lucide-react

```