



PYTHON BOOTCAMP

www.jomhack.com

FRONTEND

REACT ROUTER V7:

1. npm create vite@latest
2. Project name: . //current directory
3. Select a framework: React
4. Select a variant: React Router v7
5. Install packages: (Y)
6. Initialize a new git repository: No
7. Install dependencies with npm: Yes
8. npm run dev

Library:

- npm install lucide-react

FRONTEND

Javascript / Typescript (JS/TS):

- The interactivity of a webpage.

Cascading Style Sheets (CSS/TailwindCSS):

- Styling of a webpage

HyperText Markup Language (HTML):

- Skeleton of a webpage

References:

- <https://www.w3schools.com/>
- https://github.com/Bro-chill/fundamental_programming_languages

FRONTEND

React Hooks:

- **useState**: Manages local state
- **useEffect**: Handles lifecycle events (mount,unmount,update)
- **useRef**: Mutable reference without re-renders (Direct DOM)

React-Router Hooks:

- **useLocation**: Accesses current route information
- **useNavigate**: Navigates between routes
- **useParams**: Extracts URL parameters from route paths
- **useSearchParams**: Modifies URL query parameters
- **useRouteError**: Accesses error information

FRONTEND



React-Router Components:

- **Link:** Creates navigation link without full page reloads
- **NavLink:** Link with active state styling
- **Outlet:** Renders child route in nested routing structures
- **Form:** Form component that integrates with data features

References:

- <https://reactrouter.com/home>
- https://github.com/Bro-chill/react_router_7_fundamental

FRONTEND

react-router.config.ts:

- ssr: true (Server-Side Rendering)
- ssr: false (Single Page Application)

```
frontend > TS react-router.config.ts > ...
1  import type { Config } from "@react-router/dev/config";
2
3  export default {
4    // Config options...
5    // Server-side render by default, to enable SPA mode set this to `false`
6    ssr: true,
7  } satisfies Config;
8
```

FRONTEND



app/routes/home.tsx:

```
frontend > app > routes > ⚘ home.tsx > 📁 Home > [?] features
1  import type { Route } from "./+types/home";
2  import { Link } from "react-router";
3  import { MessageSquare, Sparkles, Zap, Shield } from "lucide-react";
4  import { AnimatedBackground } from "../components/AnimatedBackground";
5  import { FeatureCard } from "../components/FeatureCard";
6
7  export function meta({}): Route.MetaArgs {
8    return [
9      { title: "AI Assistant - Home" },
10     { name: "description", content: "Welcome to your futuristic AI assistant" },
11   ];
12 }
```

FRONTEND

app/routes/home.tsx:

```
14  export default function Home() {
15    const features = [
16      {
17        icon: Sparkles,
18        title: "Poetic Responses",
19        description: "Unique rhyming answers to all your questions",
20        gradientFrom: "from-purple-500",
21        gradientTo: "to-pink-500",
22        shadowColor: "purple-500",
23      },
24      {
25        icon: Zap,
26        title: "Lightning Fast",
27        description: "Powered by Google's Gemini AI technology",
28        gradientFrom: "from-cyan-500",
29        gradientTo: "to-blue-500",
30        shadowColor: "cyan-500",
31      },
32      {
33        icon: Shield,
34        title: "Safe & Secure",
35        description: "Content filtered for safety and appropriateness",
36        gradientFrom: "from-pink-500",
37        gradientTo: "to-purple-500",
38        shadowColor: "pink-500",
39      },
40    ];

```



FRONTEND



app/routes/home.tsx:

```
42  return (
43    <main className="relative min-h-screen flex items-center justify-center bg-gradient-to-br □from-slate-950 □via-slate-900 □to-slate-950 overflow-hidden pt-16">
44      <AnimatedBackground />
45
46      <div className="relative z-10 text-center space-y-12 p-8 max-w-4xl mx-auto">
47          {/* Logo/Icon */}
48          <div className="flex justify-center">
49              <div className="relative">
50                  <div className="absolute inset-0 bg-gradient-to-r □from-purple-500 □via-pink-500 □to-cyan-500 rounded-full blur-2xl opacity-75 animate-pulse"></div>
51                  <div className="relative bg-gradient-to-r □from-purple-500 □via-pink-500 □to-cyan-500 p-6 rounded-full shadow-2xl shadow-purple-500/50">
52                      <MessageSquare className="w-16 h-16 □text-white" />
53                  </div>
54              </div>
55          </div>
56
57          {/* Title */}
58          <div className="space-y-4">
59              <h1 className="text-6xl md:text-7xl font-bold bg-gradient-to-r □from-purple-400 □via-pink-400 □to-cyan-400 bg-clip-text text-transparent animate-fade-in">
60                  AI Assistant
61              </h1>
62              <p className="text-xl md:text-2xl □text-gray-400 font-light">
63                  Experience the future of intelligent conversation
64              </p>
65          </div>
```

FRONTEND

app/routes/home.tsx:

```
67      /* Features */
68      <div className="grid grid-cols-1 md:grid-cols-3 gap-6 mt-12">
69        {features.map((feature, index) => (
70          <FeatureCard key={index} {...feature} />
71        )));
72      </div>
73
74      /* CTA Button */
75      <div className="pt-8">
76        <Link
77          to="/chat"
78          className="group relative inline-flex items-center gap-3 px-8 py-4 bg-gradient-to-r from-purple-500 via-pink-500 to-cyan-500 rounded-2xl
79          hover:from-purple-600 hover:via-pink-600 hover:to-cyan-600
80          transition-all duration-300
81          shadow-2xl shadow-purple-500/50 hover:shadow-purple-500/75
82          text-white font-semibold text-lg"
83        >
84          <div className="absolute inset-0 bg-gradient-to-r from-purple-400 via-pink-400 to-cyan-400 rounded-2xl blur-xl opacity-0 group-hover:opacity-75 transition-opacity"></div>
85          <MessageSquare className="w-6 h-6 relative z-10" />
86          <span className="relative z-10">Start Chatting</span>
87          <Sparkles className="w-5 h-5 relative z-10 animate-pulse" />
88        </Link>
89      </div>
```

FRONTEND

app/routes/home.tsx:

```
91      /* Footer text */
92      <p className="text-gray-500 text-sm mt-12">
93          | Powered by React Router & FastAPI
94          </p>
95      </div>
96      </main>
97  );
98 }
```



FRONTEND

app/routes/chat.tsx:

```
frontend > app > routes > ✨ chat.tsx > ...
1  import { useState, useRef, useEffect } from "react";
2  import { AnimatedBackground } from "../components/AnimatedBackground";
3  import { ChatMessage } from "../components/ChatMessage";
4  import { ChatInput } from "../components/ChatInput";
5  import { LoadingIndicator } from "../components>LoadingIndicator";
6
7  interface Message {
8    role: "user" | "bot";
9    content: string;
10   timestamp: Date;
11 }
```



FRONTEND

app/routes/chat.tsx:

```
13  export function meta() {
14    return [
15      { title: "Chat - AI Assistant" },
16      { name: "description", content: "Chat with our AI assistant" },
17    ];
18  }
19
20  export default function Chat() {
21    const [messages, setMessages] = useState<Message[]>([
22      {
23        role: "bot",
24        content: "Hello! I'm your poetic AI assistant. How may I help you today?",
25        timestamp: new Date(),
26      },
27    ]);
28    const [input, setInput] = useState("");
29    const [isLoading, setIsLoading] = useState(false);
30    const messagesEndRef = useRef<HTMLDivElement>(null);
31
32    const scrollToBottom = () => {
33      messagesEndRef.current?.scrollIntoView({ behavior: "smooth" });
34    };

```



FRONTEND

app/routes/chat.tsx:

```
36  useEffect(() => {
37    scrollBottom();
38  }, [messages]);
39
40  const handleSendMessage = async () => {
41    const userMessage: Message = {
42      role: "user",
43      content: input,
44      timestamp: new Date(),
45    };
46
47    setMessages([...prev, userMessage]);
48    setInput("");
49    setIsLoading(true);
50
51    try {
52      const response = await fetch("http://localhost:8000/api/chat", {
53        method: "POST",
54        headers: {
55          "Content-Type": "application/json",
56        },
57        body: JSON.stringify({ message: input }),
58      });
59
60      if (!response.ok) {
61        throw new Error(`HTTP error! status: ${response.status}`);
62      }
63
64      const data = await response.json();
65
66      const botMessage: Message = {
67        role: "bot",
68        content: data.response,
69        timestamp: new Date(),
70      };
71
72      setMessages([...prev, botMessage]);
73    } catch (error) {
74      console.error("Error sending message:", error);
75      const errorMessage: Message = {
76        role: "bot",
77        content: "Sorry, I encountered an error. Please try again.",
78        timestamp: new Date(),
79      };
80      setMessages([...prev, errorMessage]);
81    } finally {
82      setLoading(false);
83    }
84  };

```



FRONTEND



app/routes/chat.tsx:

```
86  return (
87    <div className="relative flex flex-col h-screen bg-gradient-to-br □from-slate-950 □via-slate-900 □to-slate-950 overflow-hidden pt-16">
88      <AnimatedBackground />
89
90      {/* Messages */}
91      <div className="relative z-10 flex-1 overflow-y-auto p-6 container mx-auto max-w-5xl">
92        <div className="space-y-6">
93          {messages.map((message, index) => (
94            <ChatMessage key={index} message={message} />
95          )));
96          {isLoading && <LoadingIndicator />}
97          <div ref={messagesEndRef} />
98        </div>
99      </div>
100
101      <ChatInput
102        value={input}
103        onChange={setInput}
104        onSubmit={handleSendMessage}
105        isLoading={isLoading}
106      />
107    </div>
108  );
109}
```

FRONTEND



app/app.css:

```
frontend > app > # app.css
1  @import "tailwindcss";
2
3  @theme {
4    --font-sans: "Inter", ui-sans-serif, system-ui, sans-serif,
5    | "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
6  }
7
8  html,
9  body {
10   @apply □bg-slate-950;
11   color-scheme: dark;
12 }
13
14 /* Custom animations */
15 @keyframes fade-in {
16   from {
17     opacity: 0;
18     transform: translateY(10px);
19   }
20   to {
21     opacity: 1;
22     transform: translateY(0);
23   }
24 }
```

FRONTEND



app/app.css:

```
26  .animate-fade-in {  
27    |   animation: fade-in 0.3s ease-out;  
28  }  
29  
30  @keyframes pulse {  
31    0%, 100% {  
32      |     opacity: 0.6;  
33    }  
34    50% {  
35      |     opacity: 1;  
36    }  
37  }  
38  
39  .delay-500 {  
40    |   animation-delay: 500ms;  
41  }  
42  
43  .delay-1000 {  
44    |   animation-delay: 1000ms;  
45  }
```

```
47  /* Smooth scrollbar */  
48  ::-webkit-scrollbar {  
49    |   width: 8px;  
50  }  
51  
52  ::-webkit-scrollbar-track {  
53    |   background: rgba(15, 23, 42, 0.5);  
54  }  
55  
56  ::-webkit-scrollbar-thumb {  
57    |   background: rgba(139, 92, 246, 0.5);  
58    |   border-radius: 4px;  
59  }  
60  
61  ::-webkit-scrollbar-thumb:hover {  
62    |   background: rgba(139, 92, 246, 0.7);  
63  }
```

FRONTEND



app/app.css:

```
65  /* Glass morphism effect enhancement */
66  .backdrop-blur-xl {
67    backdrop-filter: blur(16px);
68    -webkit-backdrop-filter: blur(16px);
69  }
70
71  html,
72  body {
73    @apply ■bg-white □dark:bg-gray-950;
74
75    @media (prefers-color-scheme: dark) {
76      color-scheme: dark;
77    }
78  }
```

FRONTEND



app/routes.ts:

```
frontend > app > TS routes.ts > ...
1  import { type RouteConfig, index, route } from "@react-router/dev/routes";
2
3  export default [
4    index("routes/home.tsx"),
5    route("chat", "routes/chat.tsx")
6  ] satisfies RouteConfig;
7
```

FRONTEND



app/components/AnimatedBackground.tsx:

```
frontend > app > components >  AnimatedBackground.tsx >  AnimatedBackground
1  export function AnimatedBackground() {
2    return (
3      <div className="absolute inset-0 overflow-hidden pointer-events-none">
4        <div className="absolute -top-40 -right-40 w-80 h-80 md:w-96 md:h-96  bg-purple-500/20 rounded-full blur-3xl animate-pulse"></div>
5        <div className="absolute -bottom-40 -left-40 w-80 h-80 md:w-96 md:h-96  bg-blue-500/20 rounded-full blur-3xl animate-pulse delay-1000"></div>
6        <div className="absolute top-1/2 left-1/2 -translate-x-1/2 -translate-y-1/2 w-96 h-96 md:w-[600px] md:h-[600px]  bg-cyan-500/10 rounded-full blur-3xl animate-pulse delay-500"></div>
7      </div>
8    );
9  }
10
```

FRONTEND

app/components/ChatInput.tsx:

```
frontend > app > components > ChatInput.tsx
1 import { Send, Loader2 } from "lucide-react";
2
3 interface ChatInputProps {
4   value: string;
5   onChange: (value: string) => void;
6   onSubmit: () => void;
7   isLoading: boolean;
8 }
9
10 export function ChatInput({
11   value,
12   onChange,
13   onSubmit,
14   isLoading,
15 }: ChatInputProps) {
16   const handleSubmit = (e: React.FormEvent) => {
17     e.preventDefault();
18     if (!value.trim() || isLoading) return;
19     onSubmit();
20   };
}
```

FRONTEND



app/components/ChatInput.tsx:

```
22  return (
23    <div className="relative z-10 backdrop-blur-xl bg-white/5 border-t border-white/10 shadow-lg shadow-black/20">
24      <div className="container mx-auto max-w-5xl p-6">
25        <form onSubmit={handleSubmit} className="relative">
26          <div className="relative flex gap-3">
27            <div className="flex-1 relative group">
28              <div className="absolute inset-0 bg-gradient-to-r from-purple-500/20 to-cyan-500/20 rounded-2xl blur-xl opacity-0 group-focus-within:opacity-100 transition-opacity"></div>
29              <input
30                type="text"
31                value={value}
32                onChange={(e) => onChange(e.target.value)}
33                placeholder="Type your message..."
34                disabled={isLoading}
35                className="relative w-full px-6 py-4 bg-white/5 backdrop-blur-xl border border-white/10 rounded-2xl
36                  focus:outline-none focus:ring-2 focus:ring-purple-500/50 focus:border-purple-500/50
37                  text-gray-100 placeholder-gray-500
38                  disabled:opacity-50 disabled:cursor-not-allowed
39                  transition-all duration-300
40                  shadow-lg shadow-black/20"
41              />
42            </div>
43            <button
44              type="submit"
45              disabled={isLoading || !value.trim()}
46              className="relative group px-6 py-4 bg-gradient-to-r from-purple-500 to-cyan-500 rounded-2xl
47                  hover:from-purple-600 hover:to-cyan-600
48                  focus:outline-none focus:ring-2 focus:ring-purple-500/50
49                  disabled:opacity-50 disabled:cursor-not-allowed
50                  transition-all duration-300
51                  shadow-lg shadow-purple-500/50 hover:shadow-purple-500/75
52                  disabled:shadow-none"
53            >
```

FRONTEND



app/components/ChatInput.tsx:

```
54      <div className="absolute inset-0 bg-gradient-to-r from-purple-400 to-cyan-400 rounded-2xl blur-xl opacity-0 group-hover:opacity-75 transition-opacity"></div>
55      <div className="relative flex items-center gap-2 text-white font-medium">
56        {isLoading ? (
57          <>
58            <Loader2 className="w-5 h-5 animate-spin" />
59            <span className="hidden sm:inline">Sending</span>
60          </>
61        ) : (
62          <>
63            <Send className="w-5 h-5" />
64            <span className="hidden sm:inline">Send</span>
65          </>
66        )}
67      </div>
68      </button>
69    </div>
70  </form>
71 </div>
72 </div>
73 );
74 }
```

FRONTEND



app/components/ChatMessage.tsx:

```
frontend > app > components > ChatMessage.tsx > ...
1  import { Bot, User, Sparkles } from "lucide-react";
2
3  interface Message {
4    role: "user" | "bot";
5    content: string;
6    timestamp: Date;
7  }
8
9  interface ChatMessageProps {
10   message: Message;
11 }
12
13 export function ChatMessage({ message }: ChatMessageProps) {
14   const isBot = message.role === "bot";
15   const isUser = message.role === "user";
```

FRONTEND



app/components/ChatMessage.tsx:

```
17 |     return (
18 |       <div
19 |         className={`${`flex gap-4 ${(
20 |           isUser ? "justify-end" : "justify-start"
21 |         )} animate-fade-in`}
22 |       >
23 |         {isBot && <BotAvatar />}
24 |
25 |         <div
26 |           className={`${`group max-w-[75%] rounded-2xl backdrop-blur-xl transition-all duration-300 hover:scale-[1.02] ${(
27 |             isUser
28 |             ? "bg-gradient-to-r from-purple-500/20 to-pink-500/20 border border-purple-500/30 shadow-lg shadow-purple-500/20"
29 |             : "bg-white/5 border border-white/10 shadow-lg shadow-black/20"
30 |           )}`}
31 |         >
32 |           <div className="px-5 py-3">
33 |             <p className="text-gray-100 whitespace-pre-wrap break-words leading-relaxed">
34 |               {message.content}
35 |             </p>
36 |             <div className="flex items-center gap-2 mt-2 text-xs text-gray-400">
37 |               <span>{message.timestamp.toLocaleTimeString()}</span>
38 |               {isBot && <Sparkles className="w-3 h-3 text-cyan-400" />}
39 |             </div>
40 |           </div>
41 |         </div>
42 |
43 |         {isUser && <UserAvatar />}
44 |       </div>
45 |     );
46 |   }
```

FRONTEND



app/components/ChatMessage.tsx:

```
48 function BotAvatar() {
49   return (
50     <div className="flex-shrink-0 relative group">
51       <div className="absolute inset-0 bg-gradient-to-r from-purple-500 to-cyan-500 rounded-full blur-md opacity-0 group-hover:opacity-75 transition-opacity"></div>
52       <div className="relative w-10 h-10 rounded-full bg-gradient-to-r from-purple-500 to-cyan-500 flex items-center justify-center shadow-lg shadow-purple-500/50">
53         <Bot className="w-5 h-5 text-white" />
54       </div>
55     </div>
56   );
57 }
58
59 function UserAvatar() {
60   return (
61     <div className="flex-shrink-0 relative group">
62       <div className="absolute inset-0 bg-gradient-to-r from-pink-500 to-purple-500 rounded-full blur-md opacity-0 group-hover:opacity-75 transition-opacity"></div>
63       <div className="relative w-10 h-10 rounded-full bg-gradient-to-r from-pink-500 to-purple-500 flex items-center justify-center shadow-lg shadow-pink-500/50">
64         <User className="w-5 h-5 text-white" />
65       </div>
66     </div>
67   );
68 }
```

FRONTEND



app/components/FeatureCard.tsx:

```
frontend > app > components > FeatureCard.tsx > ...
1  import type { ComponentType, SVGProps } from "react";
2
3  interface FeatureCardProps {
4    icon: ComponentType<SVGProps<SVGSVGElement>>;
5    title: string;
6    description: string;
7    gradientFrom: string;
8    gradientTo: string;
9    shadowColor: string;
10   }
11
12  export function FeatureCard({
13    icon,
14    title,
15    description,
16    gradientFrom,
17    gradientTo,
18    shadowColor,
19  }: FeatureCardProps) {
```

FRONTEND



app/components/FeatureCard.tsx:

```
20  return (
21    <div
22      | className={`${`group backdrop-blur-xl bg-white/5 border border-white/10 rounded-2xl p-6 hover:bg-white/10 transition-all duration-300 shadow-lg shadow-black/20 hover:shadow-${shadowColor}/20`} `}
23    >
24      <div className="flex flex-col items-center gap-3">
25        <div
26          | className={`${`p-3 bg-gradient-to-r ${gradientFrom} ${gradientTo} rounded-xl shadow-lg shadow-${shadowColor}/50`} `}
27        >
28          <Icon className="w-6 h-6 text-white" />
29        </div>
30        <h3 className="text-white font-semibold">{title}</h3>
31        <p className="text-gray-400 text-sm text-center">{description}</p>
32      </div>
33    </div>
34  );
35 }
```

FRONTEND



app/components>LoadingIndicator.tsx:

```
frontend > app > components > LoadingIndicator.tsx > ...
1 import { Bot, Loader2 } from "lucide-react";
2
3 export function LoadingIndicator() {
4   return (
5     <div className="flex gap-4 justify-start animate-fade-in">
6       <div className="flex-shrink-0 w-10 h-10 rounded-full bg-gradient-to-r from-purple-500 to-cyan-500 flex items-center justify-center shadow-lg shadow-purple-500/50">
7         <Bot className="w-5 h-5 text-white" />
8       </div>
9       <div className="backdrop-blur-xl bg-white/5 border border-white/10 rounded-2xl px-5 py-3 shadow-lg shadow-black/20">
10        <div className="flex items-center gap-2">
11          <Loader2 className="w-5 h-5 animate-spin text-cyan-400" />
12          <span className="text-gray-400 text-sm">Thinking...</span>
13        </div>
14      </div>
15    </div>
16  );
17}
18
```

FRONTEND

app/components/navbar.tsx:

```
frontend > app > components > 📁 navbar.tsx > ...
1  import { Link, useLocation } from "react-router";
2  import { MessageSquare, Home, Menu, X } from "lucide-react";
3  import { useState } from "react";
4
5  export function Navbar() {
6    const location = useLocation();
7    const [isMobileMenuOpen, setIsMobileMenuOpen] = useState(false);
8
9    const navLinks = [
10      { to: "/", label: "Home", icon: Home },
11      { to: "/chat", label: "Chat", icon: MessageSquare },
12    ];
13
14    const isActive = (path: string) => location.pathname === path;
```

FRONTEND



app/components/navbar.tsx:

```
16  return (
17    <nav className="fixed top-0 left-0 right-0 z-50 backdrop-blur-xl rounded-lg border border-white shadow-lg shadow-black/20">
18      <div className="container mx-auto px-6">
19        <div className="flex items-center justify-between h-16">
20          {/* Logo */}
21          <Link to="/" className="flex items-center gap-3 group">
22            <div className="relative">
23              <div className="absolute inset-0 bg-gradient-to-r from-purple-500 to-cyan-500 rounded-full blur-lg opacity-75 group-hover:opacity-100 transition-opacity"></div>
24              <div className="relative bg-gradient-to-r from-purple-500 to-cyan-500 p-2 rounded-full">
25                <MessageSquare className="w-5 h-5 text-white" />
26              </div>
27            </div>
28            <span className="text-xl font-bold bg-gradient-to-r from-purple-400 via-pink-400 to-cyan-400 bg-clip-text text-transparent">
29              AI Assistant
30            </span>
31          </Link>
```

FRONTEND



app/components/navbar.tsx:

```
33  /* Desktop Navigation */
34  <div className="hidden md:flex items-center gap-2">
35    {navLinks.map(({ to, label, icon: Icon }) => (
36      <Link
37        key={to}
38        to={to}
39        className={`relative px-4 py-2 rounded-md transition-all duration-300
40          flex items-center gap-2 group
41          ${[
42            isActive(to)
43              ? "bg-gradient-to-r from-purple-500/20 to-cyan-500/20 text-white"
44              : "text-gray-400 hover:text-white hover:bg-white/5"
45            ]
46          }
47        `}
48      >
49        {isActive(to) && (
50          <div className="absolute inset-0 bg-gradient-to-r from-purple-500/30 to-cyan-500/30 rounded-md blur-md"></div>
51        )}
52        <Icon className="w-4 h-4 relative z-10" />
53        <span className="relative z-10 font-medium">{label}</span>
54      </Link>
55    )));
56  </div>
```

FRONTEND

app/components/navbar.tsx:

```
58      /* Mobile Menu Button */
59      <button
60          onClick={() => setIsMobileMenuOpen(!isMobileMenuOpen)}
61          className="md:hidden p-2 text-gray-400 hover:text-white transition-colors"
62          aria-label="Toggle mobile menu"
63      >
64          {isMobileMenuOpen ? (
65              <X className="w-6 h-6" />
66          ) : (
67              <Menu className="w-6 h-6" />
68          )}
69      </button>
70  </div>
```

FRONTEND

app/components/navbar.tsx:

```
72     /* Mobile Navigation */
73     {isMobileMenuOpen && (
74       <div className="md:hidden py-4 space-y-2 animate-fade-in">
75         {navLinks.map(({ to, label, icon: Icon }) => (
76           <Link
77             key={to}
78             to={to}
79             onClick={() => setIsMobileMenuOpen(false)}
80             className={`flex items-center gap-3 px-4 py-3 rounded-xl transition-all duration-300
81             ${[
82               isActive(to)
83                 ? "bg-gradient-to-r from-purple-500/20 to-cyan-500/20 text-white"
84                 : "text-gray-400 hover:text-white hover:bg-white/5"
85             ]}
86             `}
87           >
88             <Icon className="w-5 h-5" />
89             <span className="font-medium">{label}</span>
90           </Link>
91         ))})
92       </div>
93     )
94   )
95   </div>
96 </nav>
97 );
98 }
```

FRONTEND



Full Code:

- https://github.com/Bro-chill/full_stack_rag_chatbot