

Team Name: Mac Buddies

Team Members: Inchara, Kashvi G, Hassini Kunamneni

What it is ?

- Chat bot app appears to be a web-based AI chat application — essentially a “chatbot” interface you interact with in your browser.
- Its likely technical foundation is similar to what’s offered by the official Next.js AI Chatbot template from Vercel. [Vercel+1](#)
- In effect: it’s a chat UI + backend logic that lets users send messages and get AI-generated responses — much like a conversational assistant.

How it works:

- The underlying codebase for such chatbots usually uses Next.js (with its App Router + React) for the frontend. [DeepWiki+1](#)
- On the AI side, it leverages the AI SDK from Vercel, which provides a unified interface to call large-language models (LLMs), manage streaming responses, and integrate with different AI providers. [ai-sdk.dev+1](#)
- For the chat UI, the SDK exposes helpers such as the `useChat` hook (on frontend) that abstract away the complexities — so messages are handled in real time, streaming from model to user, with state management, loading/error handling, etc. [ai-sdk.dev+1](#)
- Optionally, the application may support user authentication, chat history persistence (database), file or data uploads, and even multimodal input (text, attachments) depending on configuration.

What it offers to a user:

From a user’s perspective (i.e. you — chatting with the bot):

- A clean web-UI where you can type your questions or prompts, and get AI responses instantly (streamed as they generate).
- Potential support for features beyond plain chat: depending on backend config — things like document/file uploads, context-aware replies, memory of conversation history, maybe even more advanced AI-assisted tasks.
- A lightweight, easy-to-access chat experience — no need to install software; just open in browser.

Good for:

- Quick Q&A, brainstorming, brainstorming code, writing help, general-purpose conversation — wherever a conversational AI helps.
- Trying out or building a simple AI-powered assistant without heavy infrastructure overhead (because templates + SDK + Vercel deployment manage lots of complexity).
- Web-based usage: accessible on devices, easy to share via URL