Next JS – Open AI – Chatbot

Tech Stack:

* Next JS
* Type Script
* Tailwind CSS
* OpenAI Chat Completion API

Tutorial:

1. <https://www.youtube.com/watch?v=gbZTEfo7BBo> (Done)
2. <https://www.youtube.com/watch?v=bx0oryJCpSk> (Till 54 mins)

Sequence:

1. Create Project Directory (Practiced using Git Bash)
2. Create Next JS Project (npx create-next-app@latest) with normal settings
3. Create React – Icons (npm i react-icons)
4. Created ChatBot Component
5. Created a use-effect hook with use client. (Learn more)
   1. Importing useState
   2. This is done for conditional rendering
6. Created a component for ChatBot Message.
7. Installed Open AI (npm install openai)
8. Created OpenAI Platform Account
9. Created a Secret KEY (API)
10. To Chat to OPEN AI, we have to create separate folder for Server Actions in Next.JS

Notes:

* Main page is page.tsx and to simplify the process we can delete everything in between the main tags
* Create components directory to store all components and then create the folder for separate components like ChatBot
  + Note the difference between rafc, rafce and creating components like export default function Name() { return ….. }
  + Note that the components we created needs to be called on the page.tsx to be displayed on the app.
  + Note that by default in Next JS react hooks are server components so we have to tell Next JS that it is a client component mentioning **use client**.
  + We used conditioning rendering based on the value of useState. If its sets to true it returns the divs or nothing! An arrow function alters the value of the useState so that it can create the toggle effect.
* Layout.tsx contains the meta data like title and others.
* Open AI documentation to setup is here: <https://platform.openai.com/docs/guides/chat-completions>
* See, how we created different things using the documentation like useState.
* Note the destructing done to display dynamic messages instead of hard quoted.
* Topics to be learned later (Type Script):
  + Props
  + Setting States
  + Array Maping
  + Arrow Functions
  + Destructuring
  + Spread operator …
* After setting all the functionality we talked to OpenAI
  + Saving the messages so that it can know what is the context.
  + Server actions in Next.JS
* Working with Open AI chat completion API is not that difficult. Didn’t take much to address. But it is important to understand what we have done with the documentation for better understanding.
  + For using API, payment is must!