



# AI CHAT BOT

Machine Learning (HNDIT 4212)

GAM/IT/2021/F/0036

Chalana Ravishka

## Contents

Introduction to AI Chatbots Using ChatGPT API .....	2
What is ChatGPT? .....	2
Why Use the ChatGPT API? .....	2
Prerequisites for Creating a ChatGPT-Based Chatbot .....	3
Steps to Create a ChatGPT Chatbot .....	3

# Introduction to AI Chatbots Using ChatGPT API

Artificial Intelligence (AI) has made significant strides in recent years, particularly in the field of natural language processing (NLP). One of the most exciting applications of NLP is the development of chatbots that can understand and respond to human language in a natural and engaging way. OpenAI's ChatGPT, powered by the GPT-3.5-turbo model, is a leading example of such technology.

## What is ChatGPT?

ChatGPT is a language model developed by OpenAI that uses machine learning to generate human-like text based on the input it receives. It can perform a wide range of tasks, from answering questions and providing information to engaging in conversations and assisting with various activities. The API provided by OpenAI allows developers to integrate ChatGPT into their applications, enabling them to create intelligent chatbots.

## Why Use the ChatGPT API?

Using the ChatGPT API offers several benefits:

- **High-Quality Responses:** ChatGPT generates coherent and contextually relevant responses, making interactions more natural and engaging.
- **Ease of Integration:** The API is designed to be straightforward to integrate into existing applications, whether web-based, mobile, or desktop.
- **Scalability:** OpenAI's infrastructure ensures that your chatbot can handle multiple conversations simultaneously, providing a seamless user experience.

## Prerequisites for Creating a ChatGPT-Based Chatbot

Before diving into the creation of a ChatGPT-based chatbot, a basic understanding of:

- **JavaScript and React:** React is a popular JavaScript library for building user interfaces. Familiarity with these technologies will help create a responsive and interactive chatbot.
- **API Usage:** Knowing how to make HTTP requests to interact with external APIs is crucial for sending and receiving messages from the ChatGPT API.

## Steps to Create a ChatGPT Chatbot

Here's a step-by-step guide to creating an AI chatbot using the ChatGPT API in a React application:

### Step 1: Set Up Your Environment

1. **Install Node.js and npm:** Ensure Node.js and npm installed. download them from the Node.js official website.
2. **Create a New React Application:** Use **create-react-app** to set up a new React project.

```
npx create-react-app chatgpt-chatbot  
cd chatgpt-chatbot
```

### Step 2: Install Axios

1. **Install Axios for HTTP Requests:** Axios is a library for making HTTP requests, which will use to interact with the ChatGPT API.

```
npm install axios
```

### Step 3: Obtain an API Key

1. **Sign Up for OpenAI:** Sign up on the website and obtain your API key from the API section.

#### Step 4: Create the Chat Component

1. **Create a Chat Component:** Create a new file **Chat.js** in the **src** directory.

```
1  import { useState } from "react";
2  import "./App.css";
3  import axios from "axios";
4  import ReactMarkdown from "react-markdown";
5
6  function App() {
7    const [question, setQuestion] = useState("");
8    const [answer, setAnswer] = useState("");
9    const [generatingAnswer, setGeneratingAnswer] = useState(false);
10   const [error, setError] = useState("");
11
12   async function generateAnswer(e) {
13     e.preventDefault();
14     setAnswer("");
15     setError("");
16     setGeneratingAnswer(true);
17
18     try {
19       const response = await axios.post(
20         `https://api.openai.com/v1/chat/completions?key=${import.meta.env.VITE_API_GENERATIVE_LANGUAGE_CLIENT}`,
21         {
22           contents: [{ parts: [{ text: question }] }],
23         }
24       );
25     } catch (error) {
26       setError(error.message);
27     }
28   }
29
30   return (
31     <div>
32       <input type="text" value={question} onChange={setQuestion} />
33       <button onClick={generateAnswer}>Generate Answer</button>
34       <div>
35         <div>{answer}</div>
36         <div>{error}</div>
37       </div>
38     </div>
39   );
40 }
```

```

1   setAnswer(
2     response.data.candidates[0].content.parts[0].text
3   );
4   } catch (error) {
5     console.log(error);
6     setError("Sorry - Something went wrong. Please try again!");
7   }
8   }
9   setGeneratingAnswer(false);
10  }
11
12  return (
13    <div className="bg-gradient-to-r from-green-400 to-blue-500 min-h-screen flex flex-col items-center justify-center p-4">
14      <div className="bg-white rounded-xl shadow-2xl p-8 w-full md:w-2/3 lg:w-1/2 transition-transform transform hover:scale-105">
15        <a href="https://github.com/Vishesh-Pandey/chat-ai" target="_blank" rel="noopener noreferrer">
16          <h1 className="text-4xl font-extrabold text-gray-900 mb-6 text-center animate-bounce">Smart Bot</h1>
17        </a>
18        <form onSubmit={generateAnswer} className="space-y-6">
19          <textarea
20            className="border border-gray-300 rounded w-full h-40 px-4 py-2 focus:outline-none focus:ring-4 focus:ring-green-500 transition-colors duration-300"
21            value={question}
22            onChange={e => setQuestion(e.target.value)}
23            placeholder="Ask anything"
24          />
25          <button
26            type="submit"
27            className="w-full bg-green-600 text-white px-4 py-2 rounded-md hover:bg-green-700 focus:outline-none focus:ring-4 focus:ring-green-500 transition-transform transform hover:scale-105 duration-300"
28            disabled={generatingAnswer}
29          >
30            {generatingAnswer ? "Generating..." : "Generate Answer"}
31          </button>
32        </form>

```

```

1   </form>
2   {error && <p className="text-red-500 mt-4">{error}</p>}
3   </div>
4   <div className="w-full md:w-2/3 lg:w-1/2 mt-8">
5     {answer && (
6       <div className="bg-white p-6 rounded-xl shadow-2xl transition-transform transform hover:scale-105">
7         <ReactMarkdown>{answer}</ReactMarkdown>
8       </div>
9     )}
10  </div>
11 </div>
12 );
13 }
14
15 export default App;

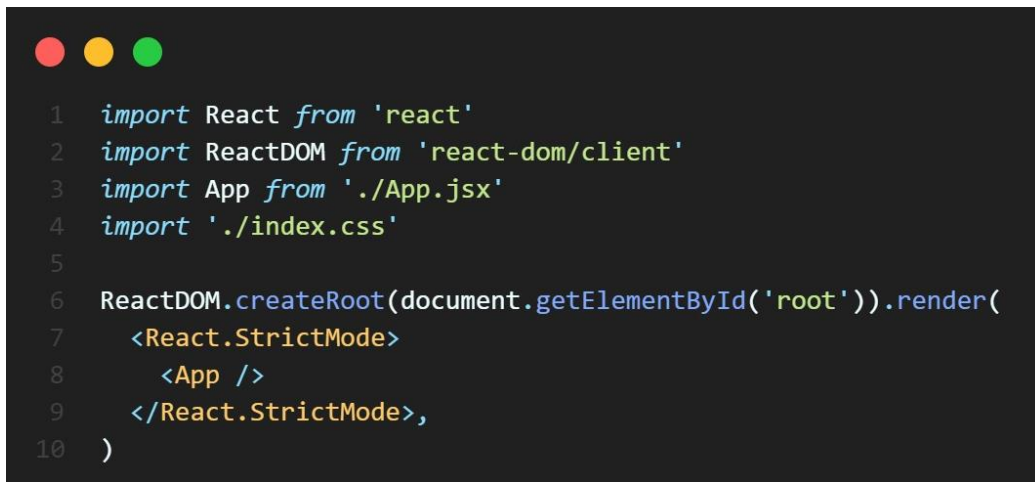
```

## Step 5: Add Basic Styling

1. **Add CSS for Chat Component:** Create a **Chat.css** file in the **src** directory.
2. **Import CSS in Chat Component:** Import the **Chat.css** file in the **Chat.js** file.

## Step 6: Integrate Chat Component in App

1. **Update App.js:** Replace the content of **App.js** to include the **Chat** component.



```
1  import React from 'react'
2  import ReactDOM from 'react-dom/client'
3  import App from './App.jsx'
4  import './index.css'
5
6  ReactDOM.createRoot(document.getElementById('root')).render(
7    <React.StrictMode>
8      <App />
9    </React.StrictMode>,
10 )
```

## Step 7: Run Your Application

Start React App: Run the following command to start your application.

```
npm run build
```