🧠 AI Integration Guide (Student Version — Full Breakdown)

In this section, we're going to walk through how to add AI to our quote generator using the OpenAI API. Every part of the code is explained in simple terms.

🔑‼️ Step 1: Set up your API Key / Retrieve Key

(For the Follow along i'll provide the API Key)

Before we do anything else, you'll need to get your own API key from OpenAI:

1. Visit https://platform.openai.com/account/api-keys
2. Click "Create new secret key"
3. Copy the key that is generated

We'll declare it in our code like this:

```
const apiKey = "your-api-key-here";
```

🔐 **Important:** Never share your real API key publicly. For this workshop, we'll include it directly in the HTML <script> section, but for real apps, you should keep it secure.

🧱 Step 2: Add Your HTML Elements (if not already present)

We assume you have this basic structure:

```
<p id="quote">Click the button to get a quote!</p>
<p id="author"></p>
<button id="new-quote">Get AI Quote</button>
```

These are the pieces we'll use to show the quote and author.

🔧 Step 3: Add the Script Tag and Variables

At the bottom of your HTML file, add a <script> tag:

```
<script>
  const apiKey = "your-api-key-here"; // Your OpenAI key

  // Grab the HTML elements
  const quoteText = document.getElementById("quote");
  const authorText = document.getElementById("author");
  const newQuote = document.getElementById("new-quote");
```

🧠 **Explanation:**

- document.getElementById("quote"): Gets the quote text element so we can change it.
- document.getElementById("author"): Gets the author element.
- document.getElementById("new-quote"): Gets the button so we know when it's clicked.

⚙️‼️ Step 4: Define the AI Quote Function

```
  async function generateQuoteFromAI() {
    quoteText.textContent = "Thinking of something inspiring..."; // Loading
message
```

- async function allows us to use await inside the function
- The quoteText.textContent changes the visible text temporarily

```
    try {
      const response = await fetch("https://api.openai.com/v1/chat/completions",
{
        method: "POST",
        headers: {
          "Content-Type": "application/json",
          "Authorization": `Bearer ${apiKey}` // Your API key
        },
        body: JSON.stringify({
          model: "gpt-3.5-turbo",
          messages: [
            { role: "system", content: "You are a helpful assistant that
generates inspiring tech-related quotes." },
            { role: "user", content: "Give me a motivational quote about
programming." }
          ],
          max_tokens: 50
        })
      });
```

- fetch is how we send the request to OpenAI's server
- method: POST means we're sending data
- headers tells the server what kind of data we're sending and includes the API key
- body includes the prompt and settings for GPT
- JSON.stringify(...) turns our object into a string that can be sent

```
      const data = await response.json(); // Convert response to usable object
      const aiQuote = data.choices[0].message.content; // Get the quote text
```

- await response.json() gets the result and converts it from JSON
- data.choices[0].message.content is where GPT puts its answer

```
      quoteText.textContent = `"${aiQuote}"`; // Show the AI quote
      authorText.textContent = "- OpenAI GPT-3.5"; // Show the source
```

```
    } catch (error) {
      quoteText.textContent = "Something went wrong. Try again.";
      console.error("Error:", error); // Show error in browser console
    }
  }
```

🧠 **Try/catch** handles errors like no internet or bad API key.

🎯 Step 5: Hook the Button to the Function

```
  newQuote.addEventListener("click", generateQuoteFromAI);
</script>
```

- This tells the browser to run generateQuoteFromAI when the button is clicked.

☑ End Result:

- A user clicks the button.
- You send a request to OpenAI.
- OpenAI sends back a motivational quote.
- The quote and author show on the screen!

Let me know if you want a challenge activity added for students or turned into a visual reference.