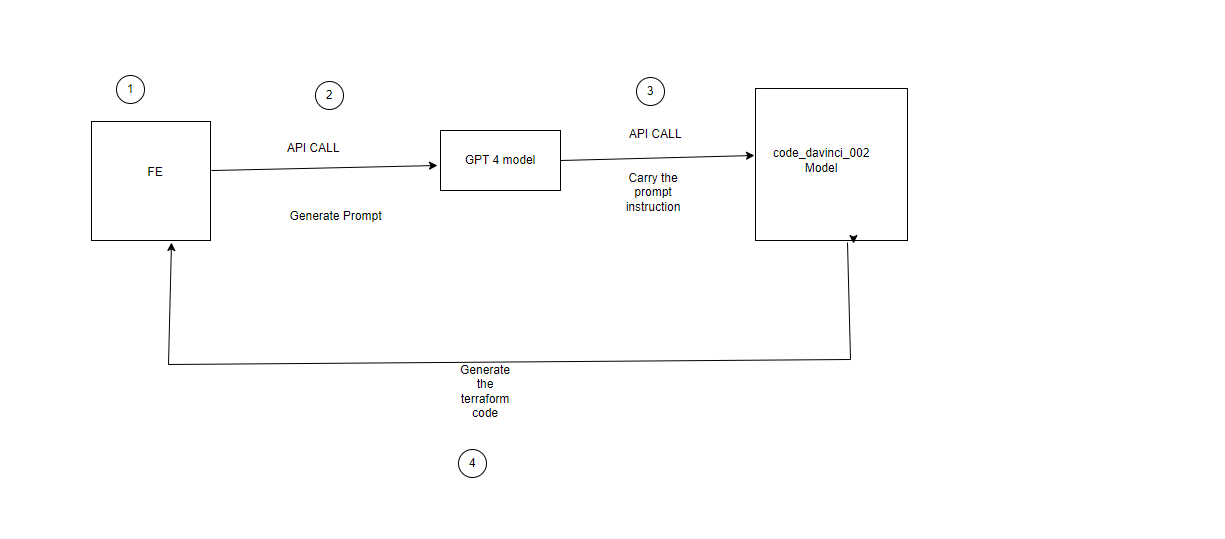
**Terraform Code Generator Using AI:**



This project involves frontend development, backend development, and integrating with Open Ai API’s like GPT 4 Model and code\_davinci\_002 model.

The Required technologies and tools to establish connection:

* Frontend Development:
  + HTML, CSS, JavaScript.
* Backend Development:
  + No Backend needed; we can execute in NodeJS and hit the GPT 3.5 model.

**PHASE 1: Creating a Front End and Integrating GPT 3.5 model API.**

Steps involved:

1. Create API key in OpenAi website and save it in a file.
2. Install VS code, node.js, Python from Browser.
3. Install Pip in local machine.
4. Check nodejs installed in local machine ( node -v)
5. Create a folder -> Move into the folder ->
6. Install Python Official bindings, “*pip install openai*”
7. To install the official OpenAI library, and express library(It manages routes and servers) run the following command in Node.js project directory:
   1. “*npm install openai@^4.0.0*”
   2. “*npm install express*”
8. Create a file in the same directory with name “server.js”, Replace the personal API key in the script and script inside is:

const express = require('express');

const OpenAI = require('openai');

const path = require('path');

const app = express();

const port = 3000;

app.use(express.json());

const openai = new OpenAI({ apiKey: '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' });

// Serve static files (CSS, JS, etc.) from the 'public' directory

app.use(express.static('public'));

app.get('/', (req, res) => {

  // Send the HTML file as the response

  res.sendFile(path.join(\_\_dirname, 'public', 'index.html'));

});

app.post('/openai-api', async (req, res) => {

  const { userMessage } = req.body;

  const response = await openai.chat.completions.create({

    model: 'gpt-3.5-turbo',

    messages: [{ role: 'user', content: userMessage }],

    temperature: 0.7,

    max\_tokens: 150,

  });

  res.json(response);

});

app.listen(port, () => {

  console.log(`Server is running at http://localhost:${port}`);

});

1. Create a folder named “public” inside the Node.js working directory, inside public folder create file names “index.html” (frontend page), “styles.css” (Styling options), “script.js”.
2. Script inside “index.html”:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="styles.css">

  <title>Modern Chat App</title>

</head>

<body>

  <div id="chat-container">

    <div id="chat-box"></div>

    <div id="user-input-container">

      <input type="text" id="user-input" placeholder="Type your message...">

      <button onclick="sendMessage()">Send</button>

    </div>

  </div>

  <script src="script.js"></script>

</body>

</html>

1. Script inside ‘styles.css’:

  body {

    font-family: 'Roboto', sans-serif;

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    background-color: #f2f2f2;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

  }

  #chat-container {

    width: 400px;

    background-color: #fff;

    border-radius: 10px;

    box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

    overflow: hidden;

  }

  #chat-box {

    height: 300px;

    overflow-y: auto;

    padding: 20px;

  }

  #user-input-container {

    display: flex;

    justify-content: space-between;

    align-items: center;

    padding: 15px;

    background-color: #f5f5f5;

  }

  #user-input {

    flex: 1;

    padding: 10px;

    border: none;

    border-radius: 5px;

    margin-right: 10px;

    box-sizing: border-box;

  }

  button {

    width: 80px;

    padding: 10px;

    background-color: #4caf50;

    color: #fff;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    box-sizing: border-box;

  }

  button:hover {

    background-color: #45a049;

  }

  button:active {

    background-color: #3e8e41;

  }

1. Script inside ‘script.js’:

const chatBox = document.getElementById('chat-box');

const userInput = document.getElementById('user-input');

async function sendMessage() {

  const userMessage = userInput.value;

  if (!userMessage) return;

  displayMessage('user', userMessage);

  const response = await getOpenAIResponse(userMessage);

  displayMessage('chatbot', response.choices[0].message.content);

  userInput.value = '';

}

function displayMessage(role, content) {

  const messageElement = document.createElement('div');

  messageElement.classList.add(role === 'user' ? 'user-message' : 'chatbot-message');

  messageElement.textContent = content;

  chatBox.appendChild(messageElement);

  chatBox.scrollTop = chatBox.scrollHeight; // Auto-scroll to the latest message

}

async function getOpenAIResponse(userMessage) {

  const response = await fetch('/openai-api', {

    method: 'POST',

    headers: {

      'Content-Type': 'application/json',

    },

    body: JSON.stringify({ userMessage }),

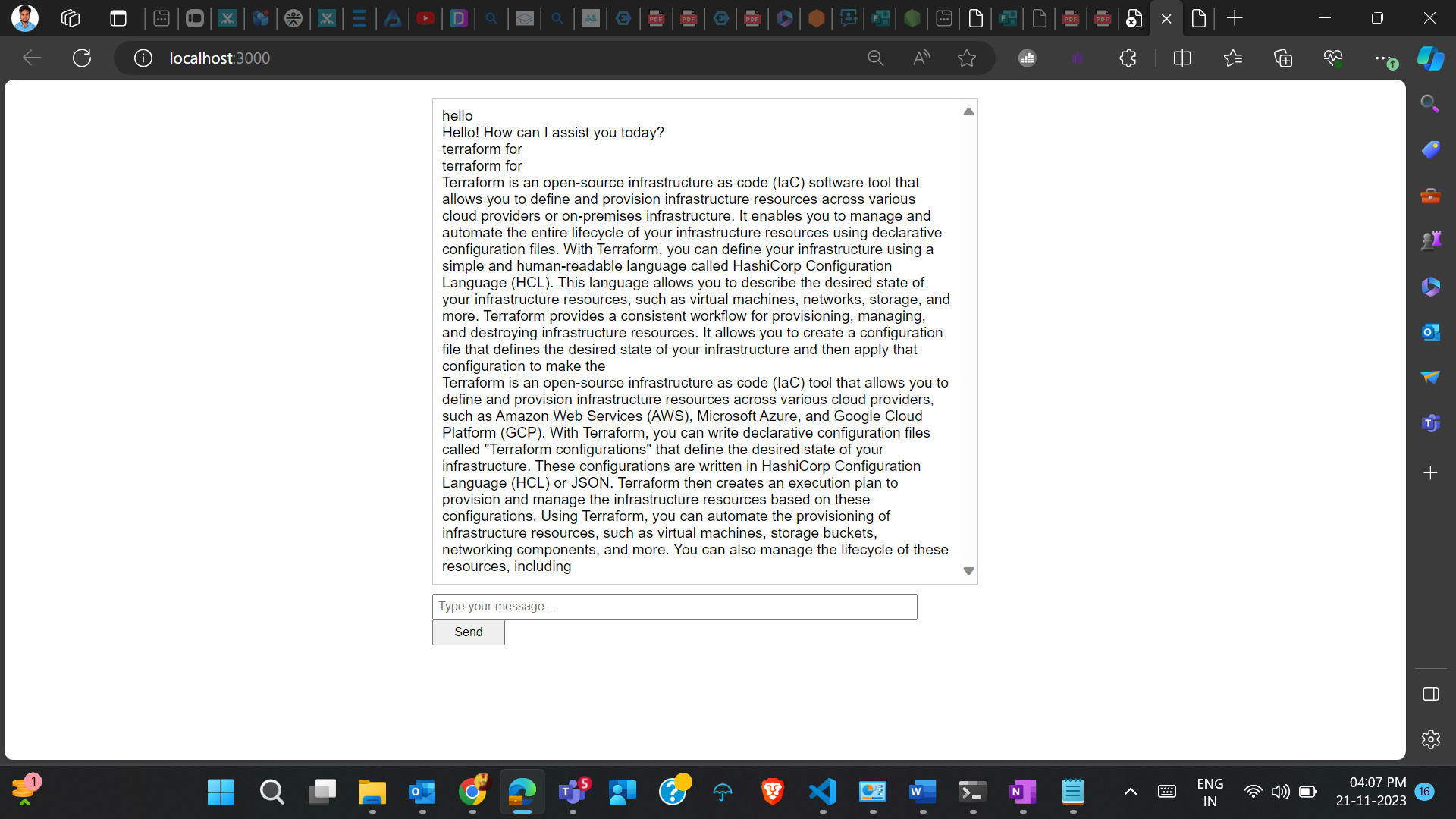
  });

  return response.json();

}

1. Execute command in main directory with command: “*node server.js*”.
2. The above command will start the engine and display the localhost address, Open the link browser to get the AI chatbot.

**Previous Output with Minimal UI:**



**Next UI Screen with Modern styles:**

