Question-1\question.txt

Question:

You have a JSON file containing question-answer pairs. Each object in the JSON array represents a question with its
Example: [{"Question":"What is your name", "Answer":"Sujoy"}]
Endpoint "/get_question":

This endpoint serves random questions from the given JSON file.
When accessed, it returns a random question along with the corresponding answer, but with each character replaced b
User Input Conditions:

For each question, show underscores () representing each character in the answer. For example, if the answer is "su
Take input from the user character by character. When the user enters a character, replace the corresponding unders
If the user provides an incorrect character, allow three chances before moving to the next character.
Scoring:

If the user correctly answers a question, assign some points for each correct answer.
Endpoint "/score":

This endpoint handles the game's scoring logic.
It determines whether the user failed or won the game based on their performance.
Provide the user with feedback on their performance.
Implementation Outline:

Read JSON File: Load the JSON file containing question-answer pairs.
Express Routes:
Set up an Express server with routes for /get_question and /score.
/get_question: Select a random question from the JSON file and replace each character in the answer with an undersc
/score: Calculate the score based on the user's input and responses.
User Input Handling:
Implement logic to handle user input for each question:
Display underscores for each character in the answer.
Allow the user to input characters and update the displayed answer accordingly.
Track incorrect guesses and limit them to three chances per character.
Scoring Logic:
Assign points for each correct answer.
Determine the user's final score based on their performance.
Endpoint Responses:
Ensure that the /get_question endpoint returns a random question with underscores for each character.
Provide appropriate responses from the /score endpoint based on the user's performance.