Gemini prompts

Prompt to “gemini-pro” model

“““Create a Parson's problem based on the theme of ${theme} and the topic of ${aiQueryTopic}. The problem should ask the user to complete a task related to this theme using Python code blocks. Provide the correct code, split into blocks, ensuring that the code logically aligns with the task. Format the response as a JSON object with two properties:

prompt: a string containing the task to be solved.

codeBlocks: an array of strings, where each string is a line of the correct code, ordered to solve the problem.

Each string in the codeBlocks array should contain no newline characters. Ensure that all dictionaries in the code are on a single line. Do not include any references to Parson's problems in the prompt field. Do not include any file opening lines of code”””

- where `${theme}` and `${aiQueryTopic}` are what the user chose as the theme and category when they are in our Problem Categories page.
- We wanted to make sure Gemini only return 2 things which are 1. the question explanation, and 2. the code blocks for the parsons problems. The question explanation are just strings, and we want the code to be split into blocks with each line before returning to us. We wanted the prompt and codeBlocks to be returned in JSON format so that it is easy to handle. We did not want our codes to have new line characters or comments or any file opening lines of code which usually exists in code generated by AI.

Topics and library used

For `${aiQueryTopic}`: These were what we passed to Gemini respectively:

```
switch (topic) {
        case 'DataFrame':
            aiQueryTopic += "Dataframes using Pandas";
            break;
        case 'NMI':
            aiQueryTopic += "Normalized Mutual Information (NMI) using scikit-learn";
            break;
        case 'Sentence Splitting':
            aiQueryTopic += "Sentence splitting using nltk.sent_tokenize()";
            break;
        case 'Correlation':
            aiQueryTopic += "Correlations using Pandas";
            break;
        case 'Linear Regression':
            aiQueryTopic += "Linear regression using Scikit-learn";
            break;
        case 'Decision Tree Classifier':
            aiQueryTopic += "Decision tree classifiers using Scikit-learn";
            break;
        case 'CSV':
            aiQueryTopic += "CSV files using pandas";
            break;
        default:
            throw new Error('Please select a prexisting topic');
    }
```