**Phase I : Planning**

**Section 1:**

The dataset appears to be a collection of challenges from LeetCode, a popular tool for coding practice, particularly among those preparing for software engineering interviews. Below is a thorough description of each attribute identified in the dataset:

* **id**: Each problem in the dataset is uniquely identified by an ID, which is a numerical value. This ID serves as a reference point, ensuring that each problem can be distinguished from others, facilitating organization and management.

Data Type: Ratio

Domain: Positive integers (e.g., 1, 2, 3, ...)

* **page\_number**: This attribute indicates the page number where a problem is located within the dataset or on the LeetCode website. It aids in pagination, allowing users to navigate through the list of problems systematically or locate them on the website with ease.

Data Type: Ratio

Domain: Positive integers indicating the page number (e.g., 1, 2, 3, ...)

* **is\_premium**: This field is a boolean value (True/False) indicating whether a problem is exclusively available to premium members of LeetCode. Premium problems require a subscription for access, thus helping users identify which problems they can attempt without a subscription.

Data Type: Categorical (Binary)

Domain: {True, False}

* **title**: Each problem is accompanied by a title, which typically includes a numerical identifier and a concise description of the challenge. The title provides a quick overview of the problem's theme or focus, assisting users in deciding if it aligns with their interests or capabilities.

Data Type: Categorical (Nominal)

Domain: Set of strings, each representing the title of a LeetCode problem (e.g., "1. Two Sum", "2. Add Two Numbers", ...)

* **problem\_description**: This attribute furnishes a comprehensive description of the coding problem, encompassing the scenario, requirements, and constraints. It is essential for understanding the problem's objectives, input/output specifications, and any specific conditions that must be met for a solution to be deemed correct.

Data Type: Categorical (Nominal)

Domain: Set of strings, with each string providing a detailed problem statement

* **topic\_tags**: These are like labels that group the problem into different areas of computer science, such as 'Array', 'String', or 'Dynamic Programming'. They help users find problems related to specific concepts or algorithms, making it easier to focus on areas they want to practice or improve.

Data Type: Categorical (Nominal)

Domain: Set of strings, each listing the topic tags associated with a problem (e.g., "'Array', 'Hash Table'", "'Linked List', 'Math', 'Recursion'", ...)

* **difficulty**: It shows how hard or easy the problem is, usually labeled as Easy, Medium, or Hard. This helps users choose problems that match their skill level or push themselves to tackle more challenging ones.

Data Type: Ordinal

Domain: {Easy, Medium, Hard}

* **similar\_questions**: This lists other problems that are similar to the current one, either in concept or difficulty. It's handy for users who want to practice more problems like the one they just solved, to strengthen their skills.

Data Type: Categorical (Nominal)

Domain: Set of strings, with each string listing similar questions to the current problem

* **no\_similar\_questions**: This number tells you how many other similar questions are available. It gives users an idea of how many more practice problems they can find related to a particular topic.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...)

* **acceptance**: The acceptance rate shows the percentage of successful submissions out of all submissions for this problem. It gives an idea of how difficult the problem is for the community.

Data Type: Ratio

Domain: Real numbers between 0 and 100 (inclusive), representing a percentage

* **accepted**: This tells you how many times the problem has been successfully solved and accepted by users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the count of accepted submissions

* **submission**: The total number of attempts made to solve the problem, whether successful or not. It indicates how popular or engaging the problem is among users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of submissions

* **solution**: It shows if there are official solutions or discussions available to help understand the problem's solution. This could be a number indicating the available solutions.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), indicating the number of solutions or solution articles available

* **discussion\_count**: This tells you how many discussions are available for the problem, including hints, solutions, and general advice from users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), indicating the number of discussion threads available

* **likes**: The total number of likes the problem has received from users, indicating its popularity or the quality of the problem statement.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of likes a problem has received

* **dislikes**: The total number of dislikes, which can highlight issues with the problem statement, its difficulty level, or user dissatisfaction.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of dislikes a problem has received

* **problem\_URL**: A direct link to the problem's page on the LeetCode website, where users can read the problem statement, submit solutions, and join discussions.

Data Type: Categorical (Nominal)

Domain: Set of URLs, each directing to the LeetCode page of a specific problem

* **solution\_URL**: Similar to the problem URL, this links directly to the official solution or a page with solutions and discussions about the problem, helping users learn and understand.

Data Type: Categorical (Nominal)

Domain: Set of URLs, each directing to the solution page on LeetCode for a specific problem

Each of these attributes gives important details about coding problems on LeetCode, making it easier for users to practice and master algorithmic challenges in a structured way.

**Section 2:**

**Recruiters:**

Recruiters and HR professionals have the opportunity to utilize the dashboard in order to assess the coding skills of candidates. This can be achieved by analyzing their performance on LeetCode problems. By doing so, recruiters can easily identify the top performers and evaluate their problem-solving abilities. Additionally, the dashboard allows them to track the trends in coding proficiency over a period of time.

**Educators:**

Educators and trainers have the ability to utilize the dashboard to track the progress of their students and pinpoint any areas that may require additional support or instruction. Furthermore, they can evaluate the efficiency of their teaching methods and curricula by analyzing the performance data of their students.

**Programmers:**

Competitive programmers or enthusiasts can utilize the dashboard to effectively monitor their rankings, identify patterns in problem-solving, and pinpoint areas for enhancing their skills. By analyzing trends in competition participation, success rates in various contest categories, and performance relative to their peers, they can gain valuable insights.

**Data analysts:**

Data analysts and researchers have the opportunity to delve into the dataset and discover valuable insights regarding coding trends, popular problem topics, and the relationships between various factors like submission frequency and problem difficulty. By utilizing the dashboard, they can effectively visualize patterns in user behavior, submission outcomes, and the progression of problem difficulty over time.

**Software Engineers:**

Software engineers and developers have the ability to utilize the dashboard to monitor their individual advancement on LeetCode. This includes visualizing various metrics like the number of problems solved over time, the distribution of difficulty levels, the accuracy rate, and their performance in specific problem categories such as algorithms and data structures. Additionally, they can compare their progress with their peers or industry benchmarks, enabling them to gain valuable insights.

**LeetCode Platform Administrators:**

The dashboard provided by the LeetCode platform enables administrators to effectively monitor user engagement, track platform usage metrics, and pinpoint areas that require improvement. By analyzing user activity patterns, administrators can identify popular features and evaluate the impact of updates or changes made to the platform.

**Academic Scholars:**

Academic scholars specializing in computer science education and learning analytics have the opportunity to utilize the dashboard for investigating problem-solving techniques, learning patterns, and the influence of online platforms such as LeetCode on the enhancement of programming skills.

**Educational Institutions:**

Educational establishments providing computer science courses have the ability to utilize the dashboard for assessing the efficiency of their curriculum, pinpointing areas that may need additional focus, and monitoring the progress of students in coding-related classes.

**Technical Interview Coaches:**

Technical interview coaches have the ability to utilize the dashboard to assess their clients' performance on LeetCode problems, pinpoint areas of strength and areas for improvement, and customize coaching sessions to meet individual needs.

**Section 3 :**

For the LeetCode dashboard designed for recruiters, educators, programmers, data analysts, software engineers, platform administrators, academic scholars, educational institutions, and technical interview coaches, user requirements might include:

* **Interactive Filters**: To sort and view data by difficulty, topic tags, acceptance rate, etc.
* **Performance Metrics Visualization**: Graphs showing problem-solving trends, acceptance rates over time, and comparison among users or groups.
* **User Engagement Analysis**: Insights into likes, dislikes, and discussion counts for identifying popular or challenging problems.
* **Ranking and Benchmarking**: For competitive programmers to compare against peers or benchmarks.
* **Progress Tracking**: For individual users to monitor their performance improvement.
* **Curriculum Efficiency**: For educators and institutions to assess curriculum impact on problem-solving skills.
* **Recruitment Tools**: To identify top performers based on specific problem categories or overall scores.
* **Security and Privacy**: Ensuring user data is protected and complies with relevant regulations.
* **Customizable Reports**: For detailed analysis tailored to specific needs of recruiters, educators, or coaches.
* **Real-time Updates**: To reflect the latest submissions, discussions, and problem updates.

Given the dataset's structure and the diverse audience it caters to, here's a list of questions that users might ask to extract valuable insights:

1. What are the most common topics/tags associated with high difficulty problems?
2. How does the number of likes/dislikes correlate with problem difficulty?
3. Which problems have the highest discussion counts, and what does this indicate about their complexity or interest?
4. Is there a trend in the number of submissions or acceptance rates over time for problems of different difficulty levels?
5. How are problem-solving trends varying across different programming topics over recent years?
6. What are the characteristics of problems with a high number of similar questions?
7. Which problems show a significant disparity between submissions and acceptance rates, indicating higher challenge levels?
8. For educators: How does student performance on certain topic tags correlate with their overall coding proficiency?
9. For recruiters: Are there specific problem tags or difficulty levels that best predict a candidate’s coding ability?
10. For platform administrators: What insights can be drawn from the relationship between problem difficulty and user engagement (likes, discussions)?
11. How does the distribution of problem difficulty levels vary across different programming topics?
12. What patterns exist in user problem-solving approaches based on the similarity of questions tackled?

These questions delve into the dataset from multiple angles, catering to the informational needs of recruiters, educators, programmers, data analysts, software engineers, platform administrators, academic scholars, educational institutions, and technical interview coaches.

**Section 4 :**

Link to the dataset - <https://www.kaggle.com/datasets/jaydeepagravat94583/leetcode>

Link to Mural - <https://app.mural.co/t/team605123/m/team605123/1712279236151/f7b58b853b8797e47d29e720117ffea18007cc61?sender=u989641d48ac1c05fb15d3600>