

# FINAL PRESENTATION

Presented by Team AskMe

# TEAM MEMBERS

**MD Akib Haider**  
**210041222**

**Sameen Yeaser**  
**210041234**

**Sakeef Hossain**  
**210041220**

**Faiyaz Abrar**  
**210041214**

01

# PROJECT OVERVIEW

Next generation coding platform  
convenient for all type of coders as well  
as new learners.

# FEATURES

Adding  
Problems

Filtering  
Problems

Code  
Compilation

User Note

Authentication

# ADDING PROBLEMS

03

## Create Problem

Get started by providing the initial details needed to create a problem.

Problem Category

Problem Name

Problem Statement

**B** *I* S HR TITL | ♂ ” ↔ ≡ ⊕ ⊗ ⊕ ⊗ | ?  
[CC HC | HC]

Input Format

**B** *I* S HR TITL | ♂ ” ↔ ≡ ⊕ ⊗ ⊕ ⊗ | ?  
[CC HC | HC]

Add Testcase

Proceed

01

Specify Problem Category for the new problem

02

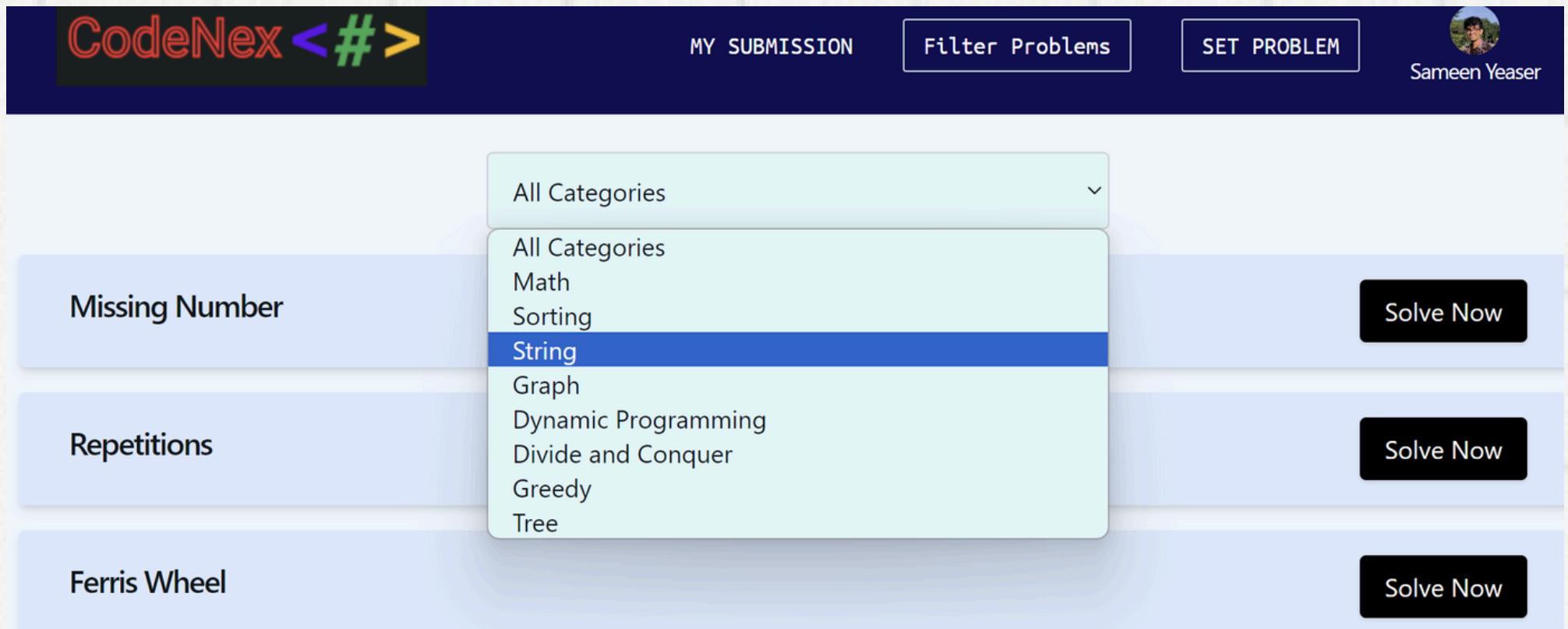
Add the test case along with creating the problem

03

Specify the constraints, input format and output format while creation of the problem.

# FILTERING PROBLEMS

04



The screenshot shows a user interface for a programming challenge platform. At the top, there is a navigation bar with the logo 'CodeNex <#>', 'MY SUBMISSION', 'Filter Problems' (which is the active tab), 'SET PROBLEM', and a user profile for 'Sameen Yeaser'. Below the navigation bar, there is a list of problems with their titles and a 'Solve Now' button. The first problem listed is 'Missing Number'. To the right of the problem titles is a dropdown menu titled 'All Categories' which lists various algorithmic categories: All Categories, Math, Sorting, String, Graph, Dynamic Programming, Divide and Conquer, Greedy, and Tree. The 'String' category is currently selected, as indicated by a blue underline. The 'Solve Now' button for the 'Missing Number' problem is also highlighted with a blue underline.

Problem	Solve Now
Missing Number	Solve Now
Repetitions	Solve Now
Ferris Wheel	Solve Now

User can filter problems based on different categories and view desired problems.

# CODE COMPILATION

05

## Description

### Problem Statement

You are given all numbers between 1,2,...,n except one. Your task is to find the missing number.

Input The first input line contains an integer n. The second line contains n-1 numbers. Each number is distinct and between 1 and n (inclusive).

### Output Format

Print one integer

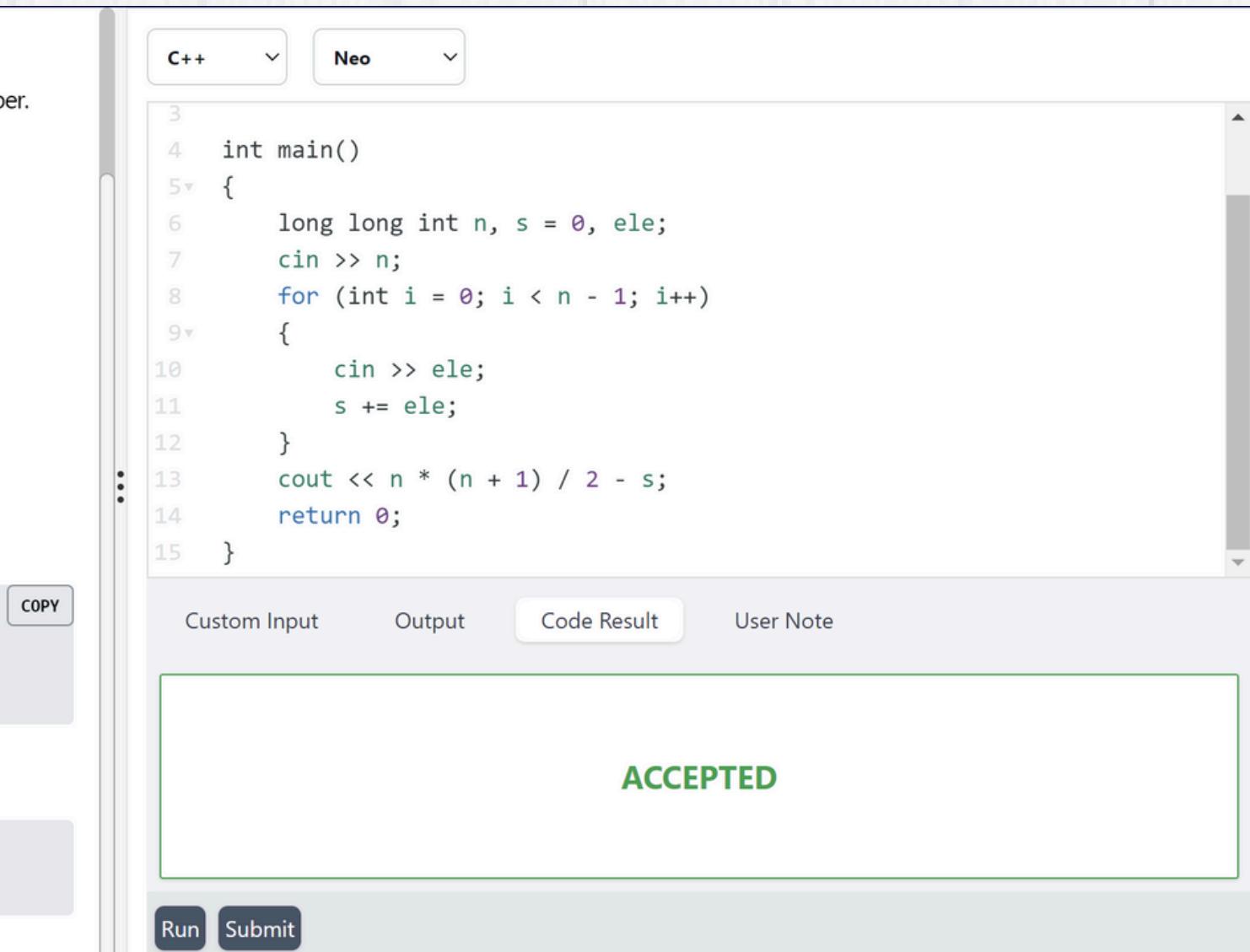
### Constraints

$1 < n < 2e5 + 1$

### Sample Input 1

```
4
```

```
1 3 4
```



```
3
4 int main()
5 {
6     long long int n, s = 0, ele;
7     cin >> n;
8     for (int i = 0; i < n - 1; i++)
9     {
10         cin >> ele;
11         s += ele;
12     }
13     cout << n * (n + 1) / 2 - s;
14     return 0;
15 }
```

COPY

Custom Input   Output   **Code Result**   User Note

**ACCEPTED**

Run   Submit

### Sample Output 1

```
2
```

## Code Editor

Provide code solution obtained from the user

## Run Code

Compile and run the code along with custom input for the code provided.

## Code Submission

Submission of code and view the submission status if it was accepted or not

# USER NOTE

06

Custom Input

Output

Code Result

User Note

Three variables were used.|

Save Note

Run Submit

User can add notes to their submission to add any comments on their codes.

# HIGH LEVEL IMPLEMENTATION

07

## User Authentication

Use of OAuth 2.0 for secure login (e.g., Google, GitHub). Implement role-based access control to differentiate between regular users and problem setters.

## Problem Management

Implement backend APIs to handle CRUD (Create, Read, Update, Delete) operations on problems. Ensure proper validation and error handling during problem management operations.

## Problem Filtering and Search

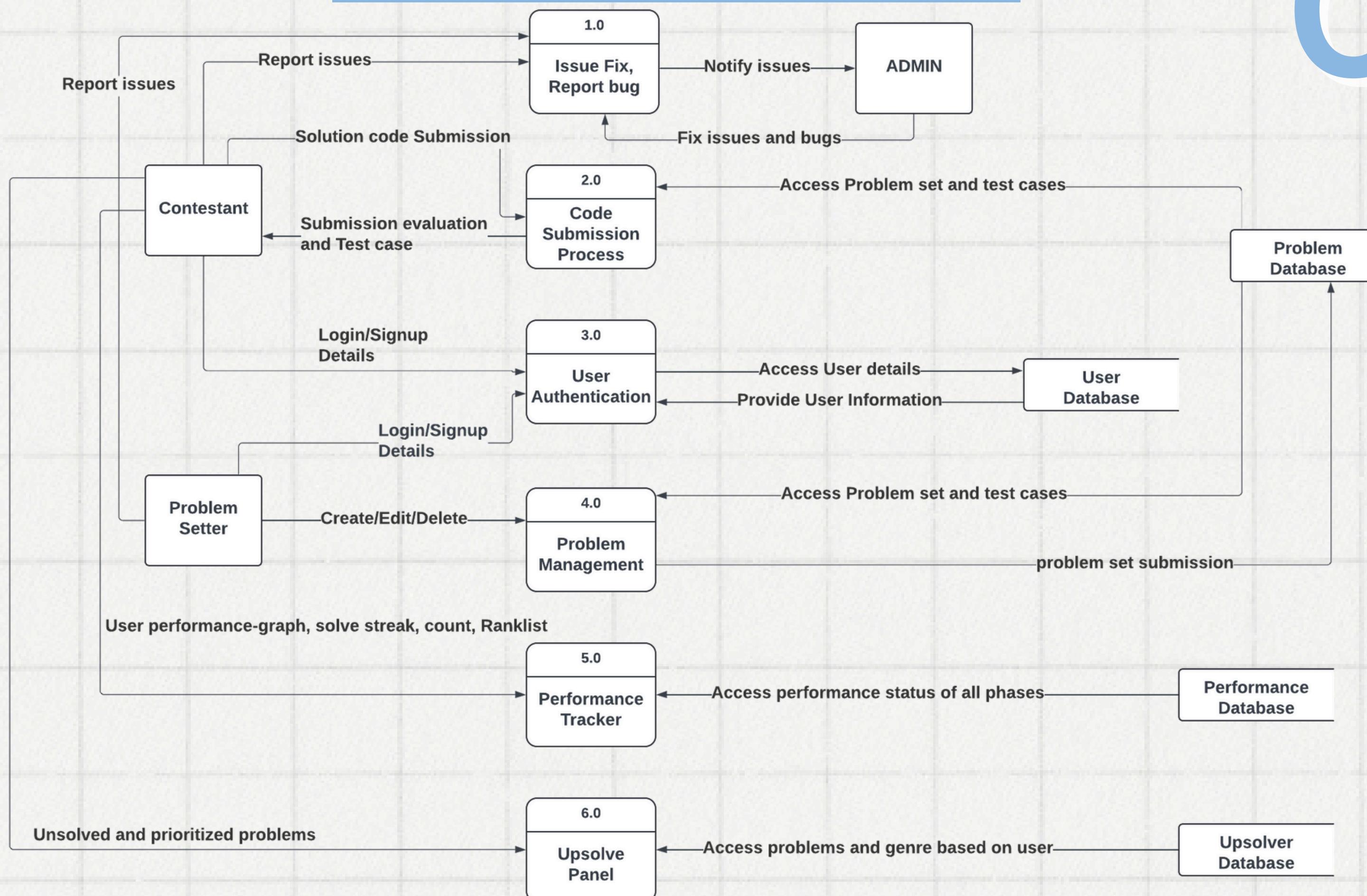
Implement advanced search functionality with filters for difficulty level, tags, and popularity. Use indexing and efficient query mechanisms to ensure fast and responsive search results.

## Scalable Backend

Compare user output with expected output and assign scores. Implement ranking and leaderboard module. Integrate with the frontend via APIs.

# DIAGRAM 0

08



**Thank you !!**