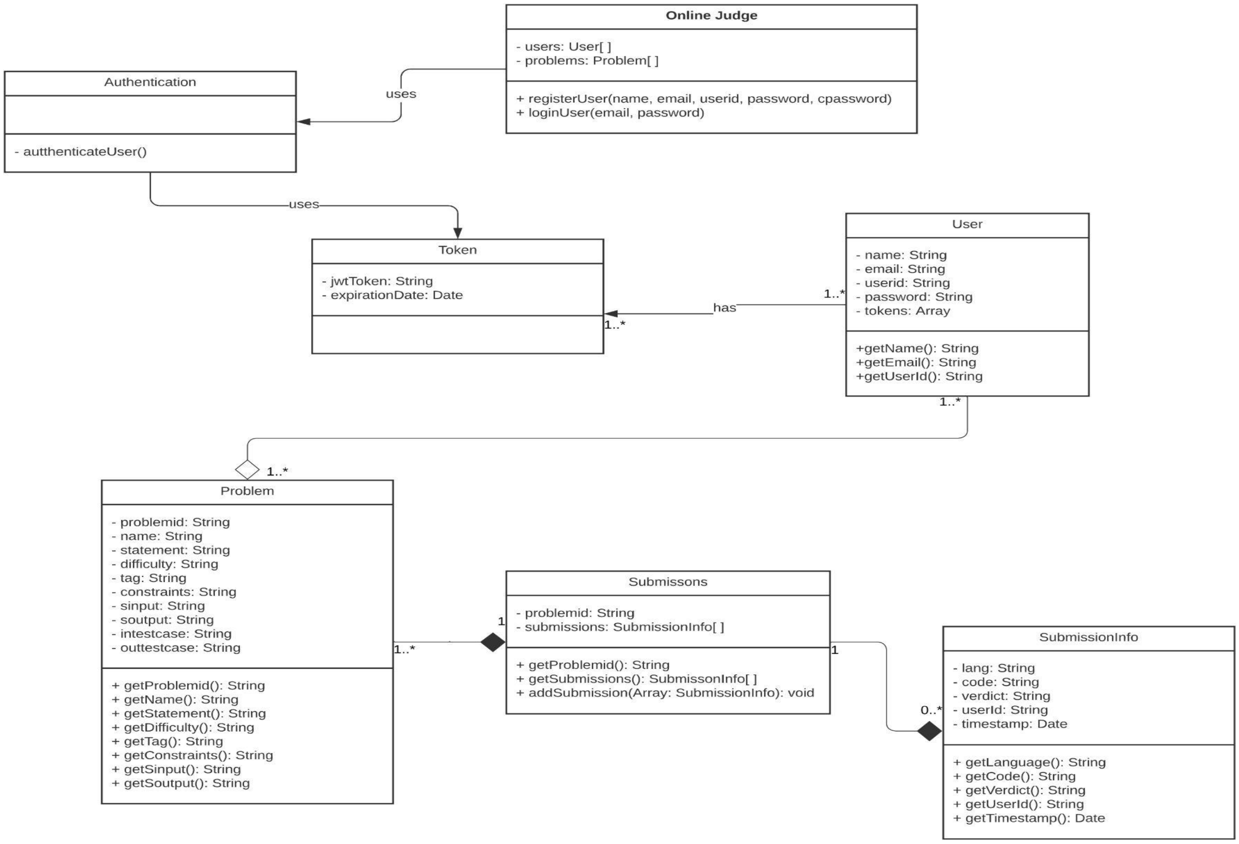
Low-Level Design Document

1. **Overview**
   The Online Judge project aims to provide a web-based platform for users to solve coding problems, submit solutions, and view the verdicts. The system supports user registration, login, and authentication using JWT tokens. Users can view and solve problems, submit their code, and see the status of their submissions.

2. **Components**
   - User Management: Responsible for user registration, login, and authentication.
   - Problem Management: Manages the collection of coding problems available on the platform.
   - Submission Management: Handles user code submissions and their verdicts.
   - Frontend Interface: Provides a user-friendly interface to interact with the system.
   - Database: Stores user information, problem details, and submissions.

3. **Class Diagram**
   The below UML class diagram forms the basis of the Online Judge project's Low-Level Design.



4. **Interaction Flow**

- User visits the Online Judge website and sees the navigation bar with options: Home, Problems, Register, and Login.
- If the user is not registered, they can click on the Register option and provide the required details (name, email, userid, password) to register.
- Once registered, the user can click on the Login option to provide their credentials. The system will authenticate the user's credentials and generate a JWT token associated with the user's account.
- The user can now access the Problems option from the navigation bar, which redirects them to the list of coding problems.
- Upon clicking on a specific problem, the user can view the problem details, including description, constraints, and sample test cases.
- The user can select a programming language (e.g., C++, Java, Python) and write their code in the provided textarea.
- After writing the code, the user can click on the Run Code button to see the output for the provided sample test cases.
- If satisfied with the code, the user can click on the Submit Code button to submit the solution.
- The system will evaluate the submission and provide a verdict (e.g., Accepted, Wrong Answer) based on the provided code.
- The user can also navigate to the Go to all Submissions page to view the past submissions of all users, including details like the language used, code written and the verdict received.

6. **Database Schema**

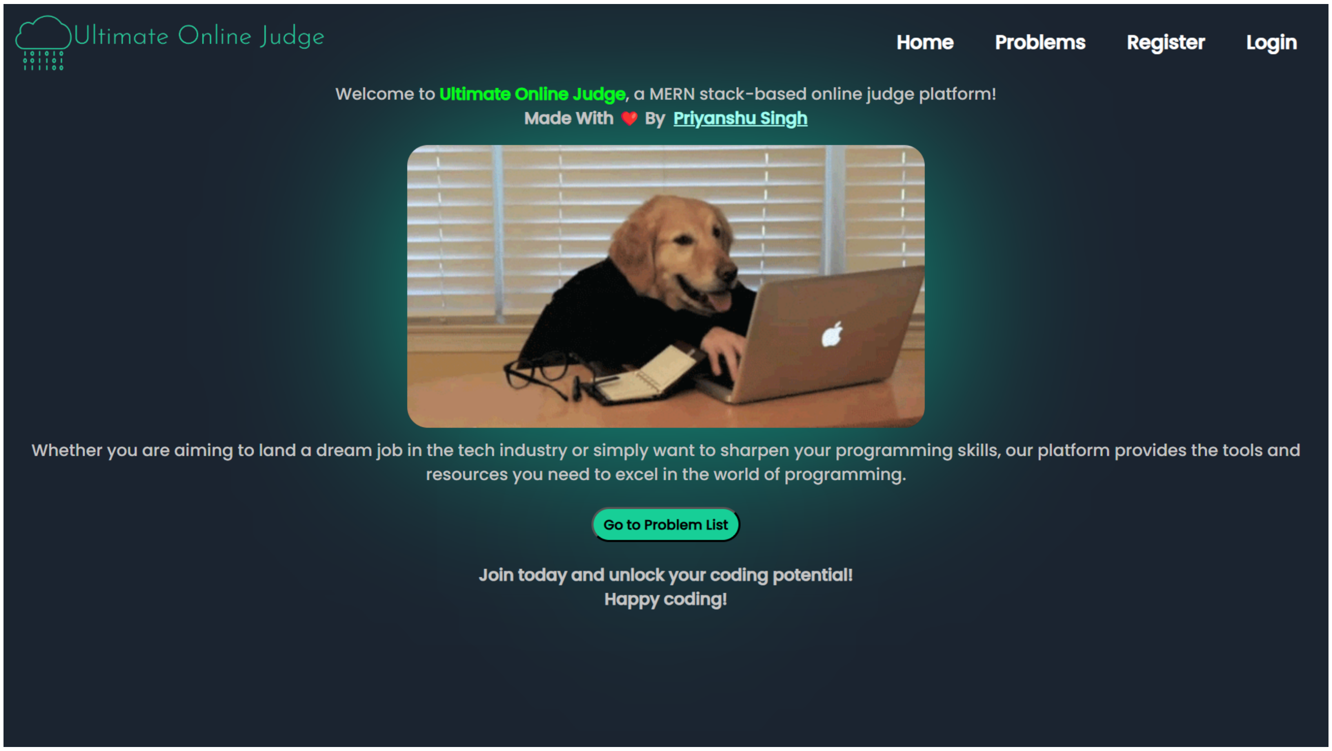
The project will require a database to store user information, problem details, and submissions. The database schema will include tables for users, problems, and submissions, with appropriate relationships and constraints as shown in the UML diagram.

7. **Conclusion**

The Low-Level Design (LLD) document outlines the structure and components of the Online Judge project. It defines the main classes, their relationships, and their roles within the system. The described functionalities, interaction flow, and database schema provide a foundation for implementing the project. The detailed implementation will require defining methods, validations, and handling various scenarios as per the project's requirements.

Some UI Screens Images

Home Page

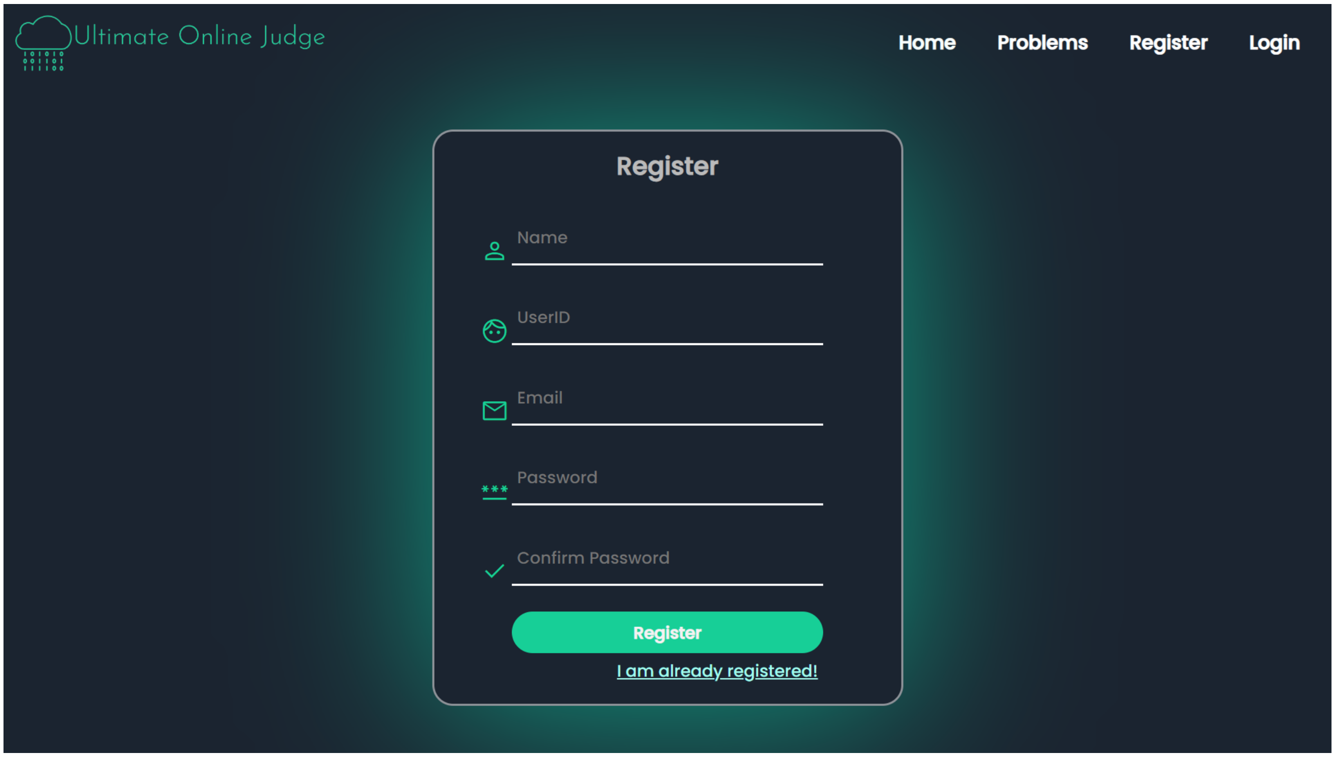


Welcome to **Ultimate Online Judge**, a MERN stack-based online judge platform!
Made With ❤ By Priyanshu Singh

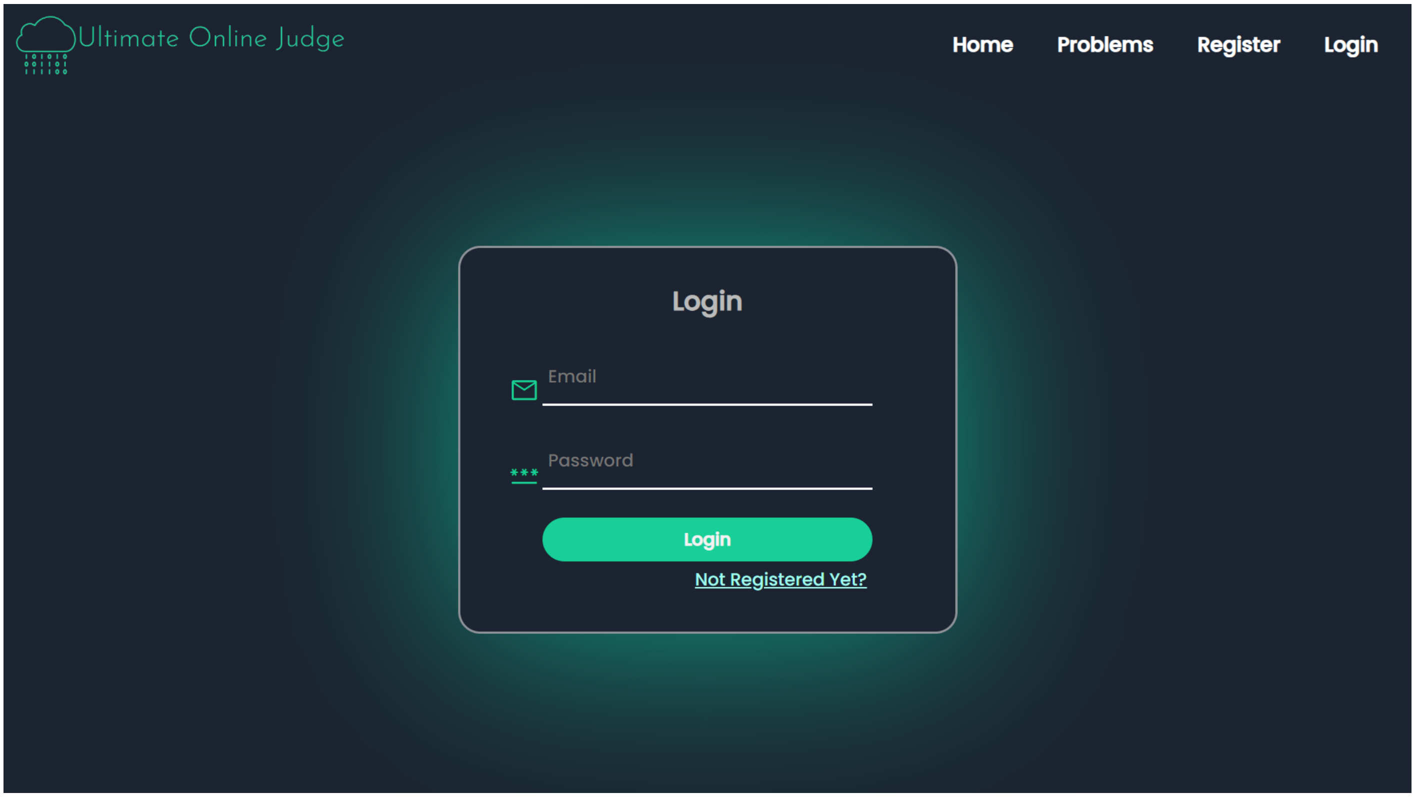
Whether you are aiming to land a dream job in the tech industry or simply want to sharpen your programming skills, our platform provides the tools and resources you need to excel in the world of programming.

Join today and unlock your coding potential!
Happy coding!

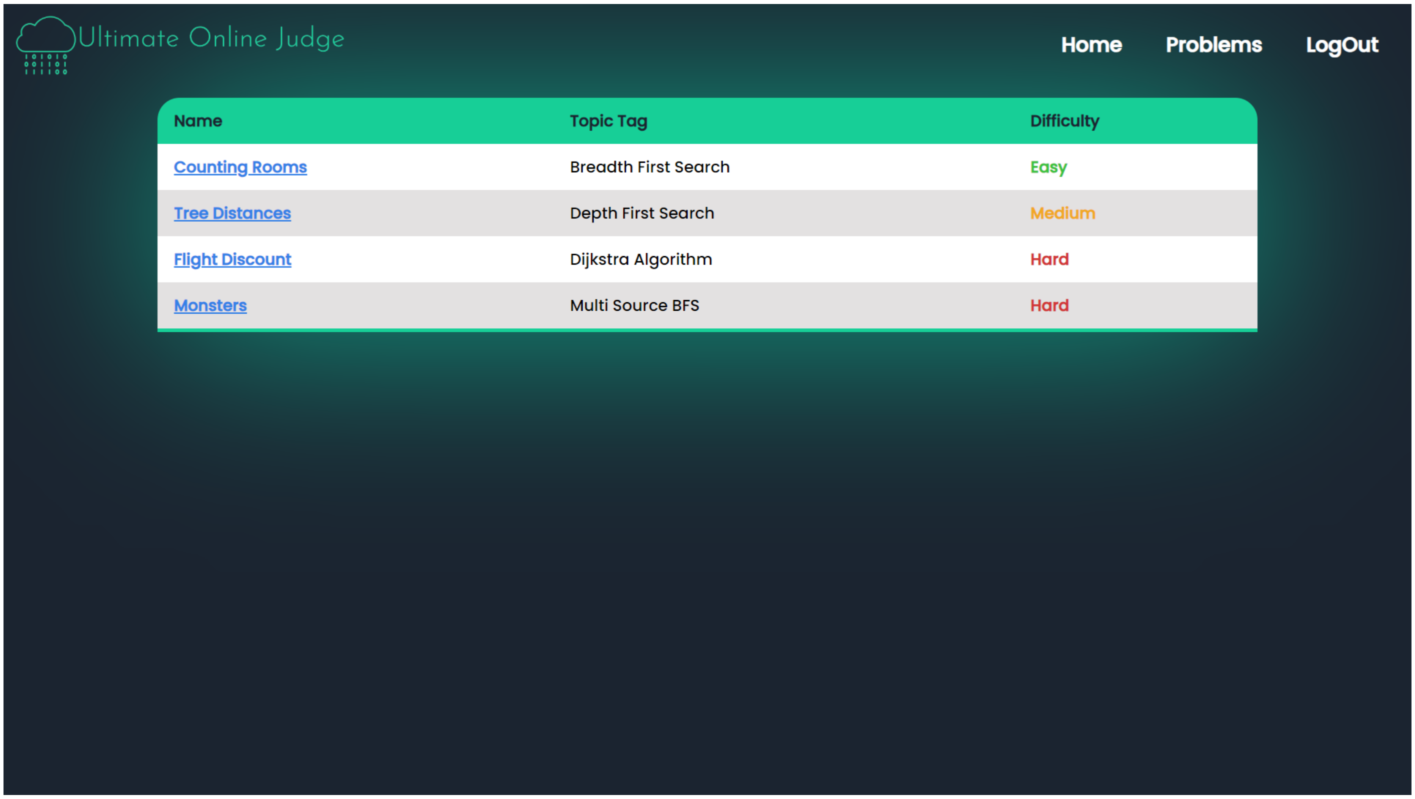
Register Page



Login Page

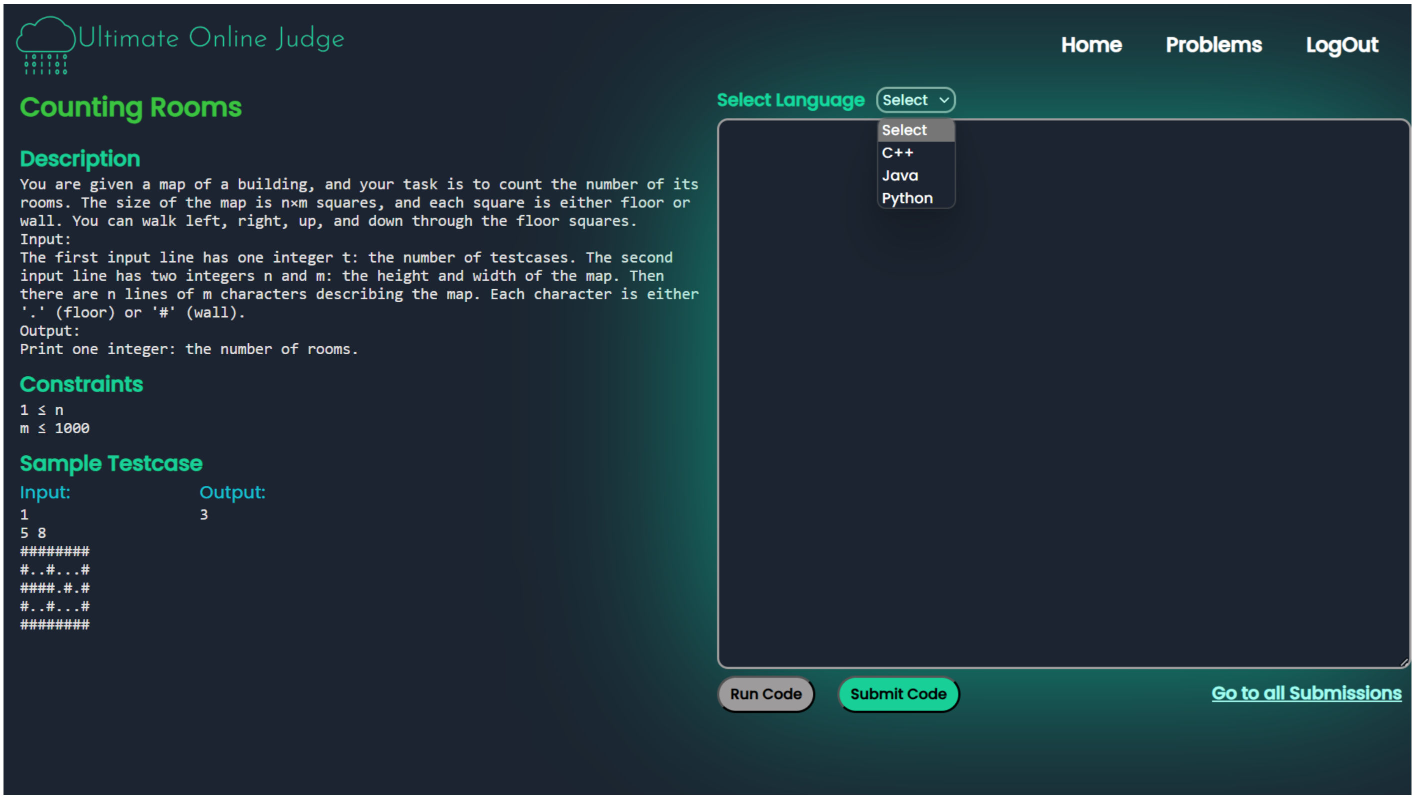


Problems Page



Name	Topic Tag	Difficulty
Counting Rooms	Breadth First Search	Easy
Tree Distances	Depth First Search	Medium
Flight Discount	Dijkstra Algorithm	Hard
Monsters	Multi Source BFS	Hard

Problem Statement Page



Counting Rooms

Description

You are given a map of a building, and your task is to count the number of its rooms. The size of the map is n×m squares, and each square is either floor or wall. You can walk left, right, up, and down through the floor squares.
Input:
The first input line has one integer t: the number of testcases. The second input line has two integers n and m: the height and width of the map. Then there are n lines of m characters describing the map. Each character is either '.' (floor) or '#' (wall).
Output:
Print one integer: the number of rooms.

Constraints

1 ≤ n
m ≤ 1000

Sample Testcase

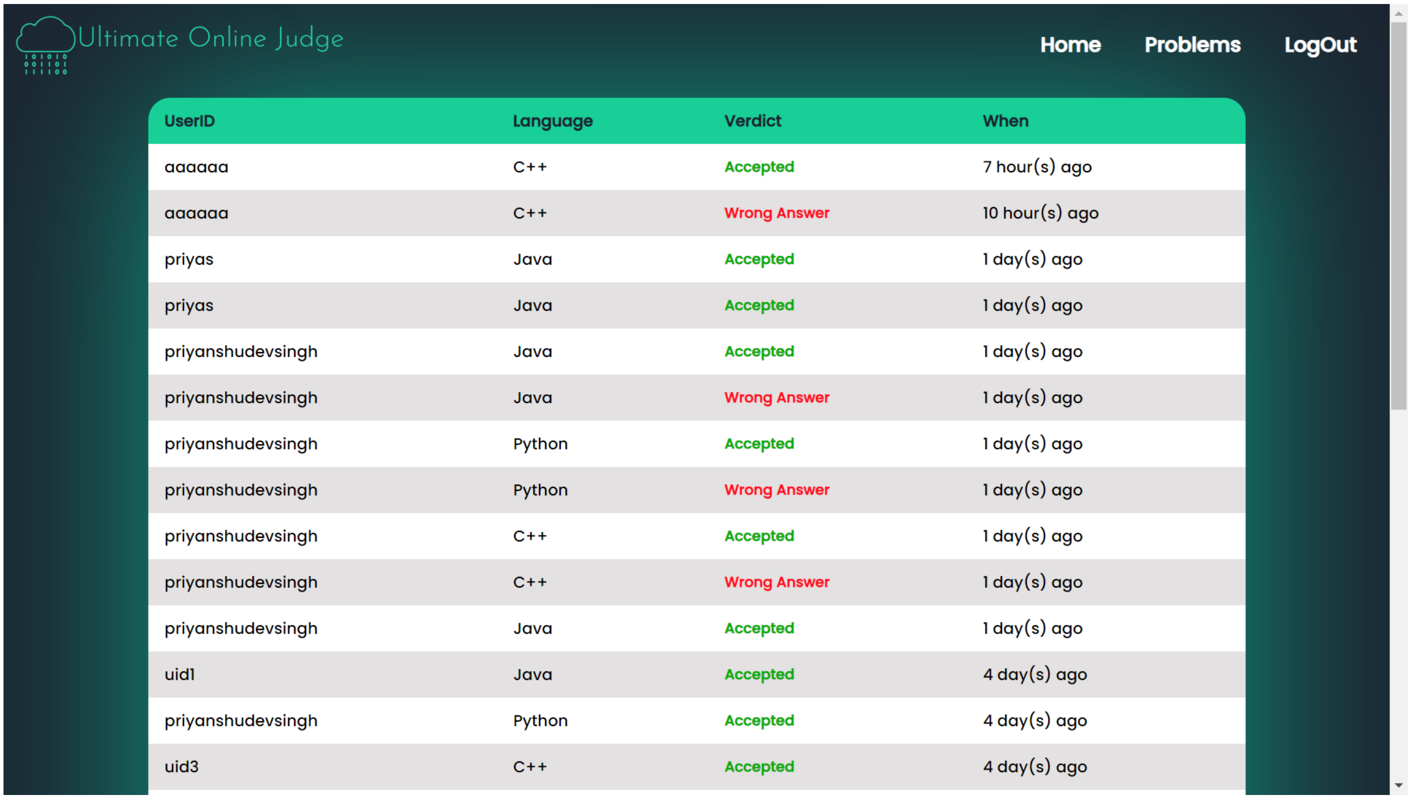
Input:

```
1
5 8
########
#..#...#
####.#.#
#..#...#
########
```

Output:

```
3
```

Submissions Page



UserID	Language	Verdict	When
aaaaaa	C++	Accepted	7 hour(s) ago
aaaaaa	C++	Wrong Answer	10 hour(s) ago
priyas	Java	Accepted	1 day(s) ago
priyas	Java	Accepted	1 day(s) ago
priyanshudevsingh	Java	Accepted	1 day(s) ago
priyanshudevsingh	Java	Wrong Answer	1 day(s) ago
priyanshudevsingh	Python	Accepted	1 day(s) ago
priyanshudevsingh	Python	Wrong Answer	1 day(s) ago
priyanshudevsingh	C++	Accepted	1 day(s) ago
priyanshudevsingh	C++	Wrong Answer	1 day(s) ago
priyanshudevsingh	Java	Accepted	1 day(s) ago
uid1	Java	Accepted	4 day(s) ago
priyanshudevsingh	Python	Accepted	4 day(s) ago
uid3	C++	Accepted	4 day(s) ago