

GROUP #1

CODEAREA: ONLINE JUDGE PLATFORM

</about>

It is a competitive programming platform for local competitions at IIT Ropar. It is developed with Django(Python based Web Development Framework).

Features:

- Problems, Contests and Posts.
- Programming Languages: C++/Python/Go/Java.

</user>

User can either sign up using our default sign up interface, where they'll receive a verification mail on their given email or can avoid that by signing in with Google

</problems>

Any user can add problems. Only they will be allowed to add or remove test cases. Time limit for each test case is 2 seconds. A problem can be unlisted so it does not appear on the feed. Problems can be filtered through tags and autocomplete search

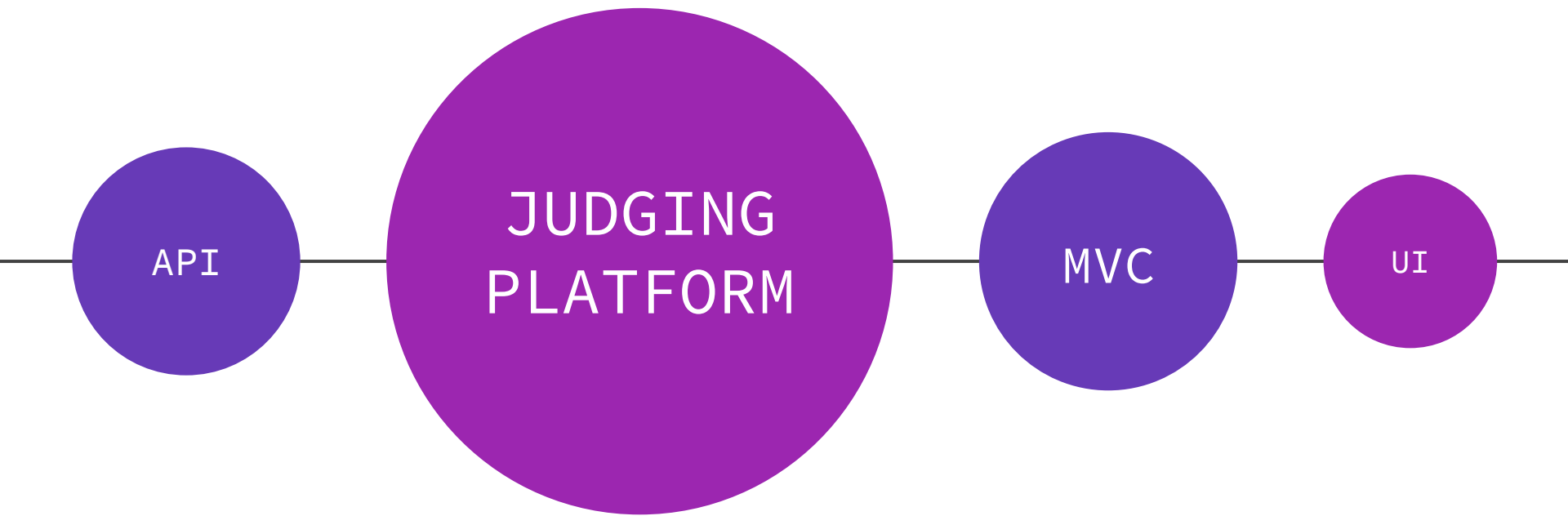
</contests>

Any user can create a contest. Contest problems would be selected from the list of problems. Users cannot see contest problems until the contest has started and when the contest ends, their scores won't be added.

</posts>

Any user can add and like posts. Posts, like problems can be filtered with tags.

`</modules>`



</tech>

BACKEND: Postgres + Django

API: Django REST Framework

IDE: ACE Editor

SANDBOXING: Docker

UI: jQuery, HTML5, CSS3, select2,
markdownjs, bootstrap, material kit

WIDGETS: Pagedown, Django AllAuth

— — —

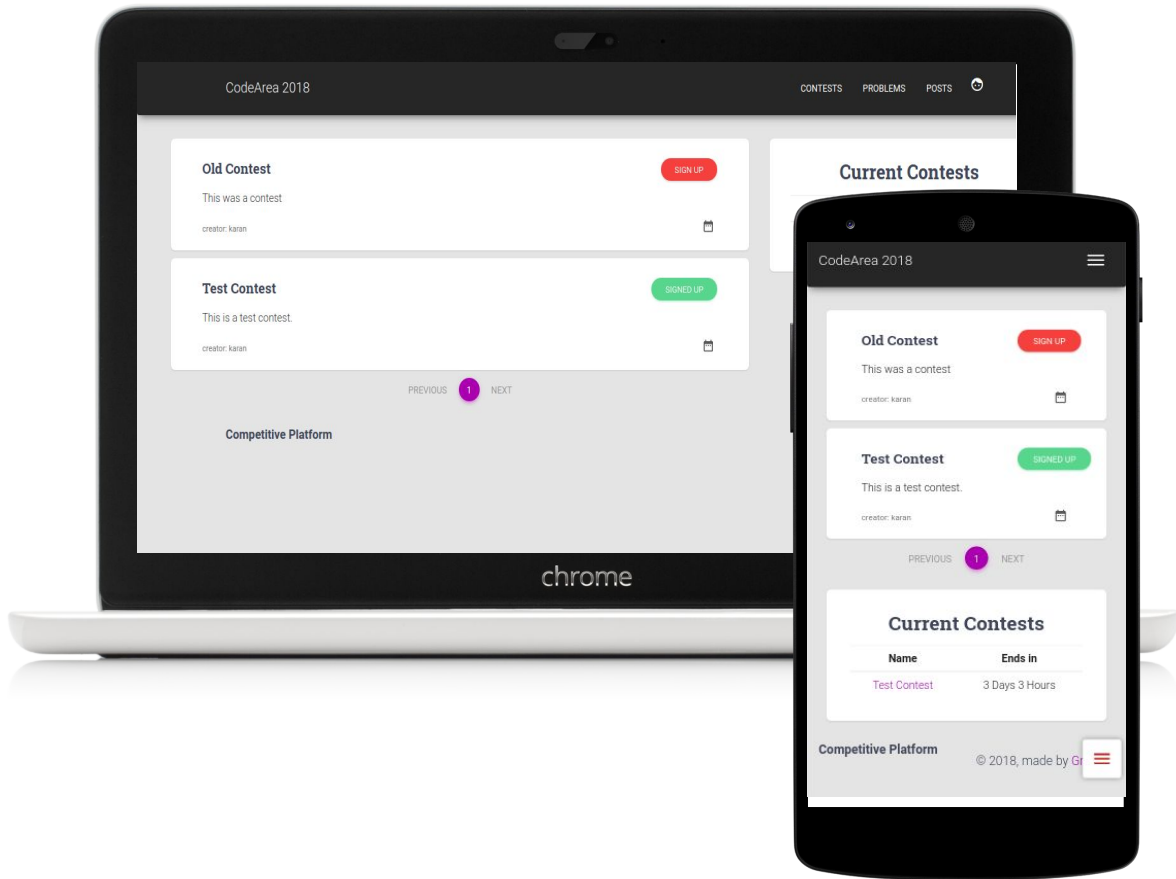
A CLOSER LOOK

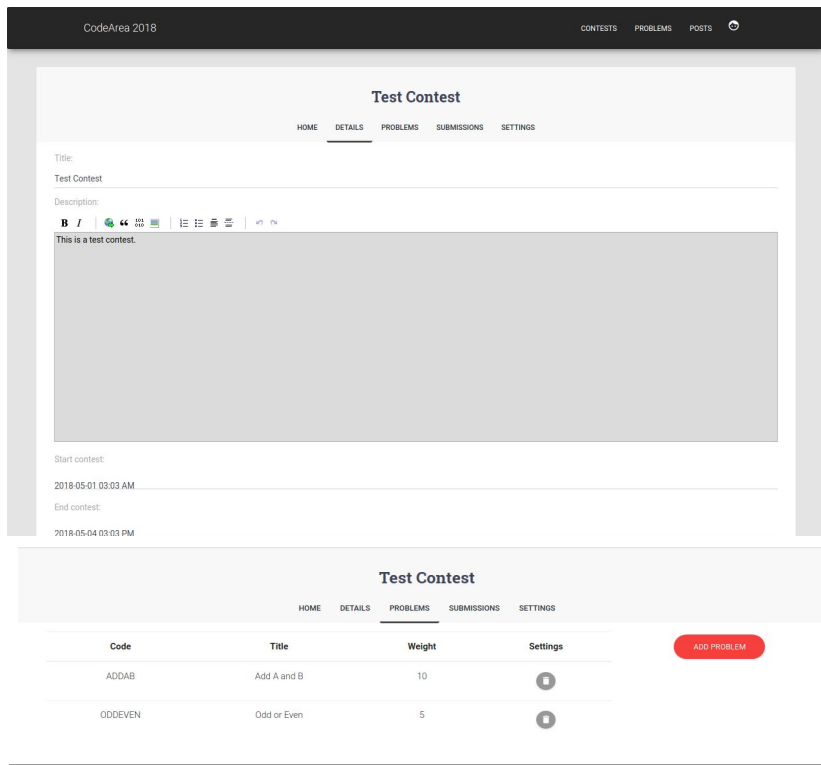
CODE AREA

— — —

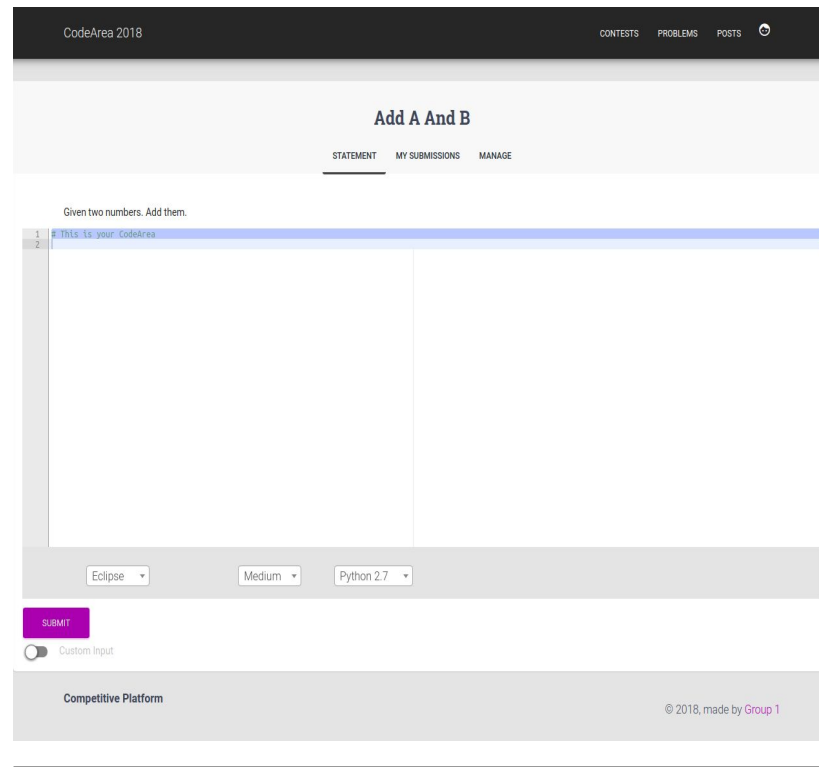
The responsive UI is made from scratch using Bootstrap 4 and a Material UI Kit.

A major part of the UI includes communication with our backend API using AJAX, to give the user an amazing experience





ADMIN INTERFACE



ACE IDE

Django REST framework

karan

Problem List

FiltersOPTIONSGET

GET /problems/api/problems/

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[
  {
    "id": 12,
    "title": "Add A and B",
    "problem_code": "ADDB",
    "statement": "Given two numbers. Add them.",
    "timestamp": "2018-05-01T09:31:46.809495+05:30",
    "level": "EASY",
    "unlisted": false,
    "setter": 1,
    "tags": [
      6
    ]
  },
  {
    "id": 11,
    "title": "Odd or Even",
    "problem_code": "ODDEVEN",
    "statement": "Given a number, print whether the number of odd or even.",
    "timestamp": "2018-05-01T09:28:09.094369+05:30",
    "level": "EASY",
    "unlisted": false,
    "setter": 1,
    "tags": [
      6
    ]
  }
]
```

Raw dataHTML form

Title

Problem code

Statement

Level

Unlisted

Setter

Tags

POST

AND MUCH MORE...

API

MVC

MODEL

Interface to our data

VIEW

How data is presented

CONTROLLER

Flow between Model and View.
Uses CSRF tokens to avoid CSRF attacks.

— — —

ONLINE JUDGE

SUPPORTS C/C++/JAVA/PYTHON/GO

USES DOCKER FOR SANDBOXING

THE JUDGE

The most important thing for online judge is security.

Because a malicious code can harm the server. And the damage can be drastic because the judge running the user's program directly on the server.

So , to isolate the user's program from the server we have used the security provided by Docker containers.

A Docker container provide the user's program a isolated environment with necessary libraries.



HOW JUDGE EXECUTE THE PROGRAM

- The judge gets the ajax request from client.
- Judge prepares a Docker container for client's program.
- Execute it with time limit of 2 sec.
- Save the result in database.
- Return an HttpResponseMessage to client.

The judge provide runtime and compile time errors to the user.

The judge three types of execution.

1. Normal Execution
2. Practice Problem Execution
3. Contest Problem Execution

```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . ", port);
        } catch (SocketException){
            Console.WriteLine("Unable to connect to server.");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData = Encoding.
            ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while(true){
            input = Console.ReadLine();
            if (input == "exit") break;
            newchild.Properties["ou"].Add
            ("Auditing Department");
            if (input == "CommitChanges"){
                newchild.CommitChanges();
                newchild.Close();
            }
        }
    }
}
```

Contributors

— — —

Karan Sehgal

(2016CSB1080)

Girish Kumar

(2016CSB1040)

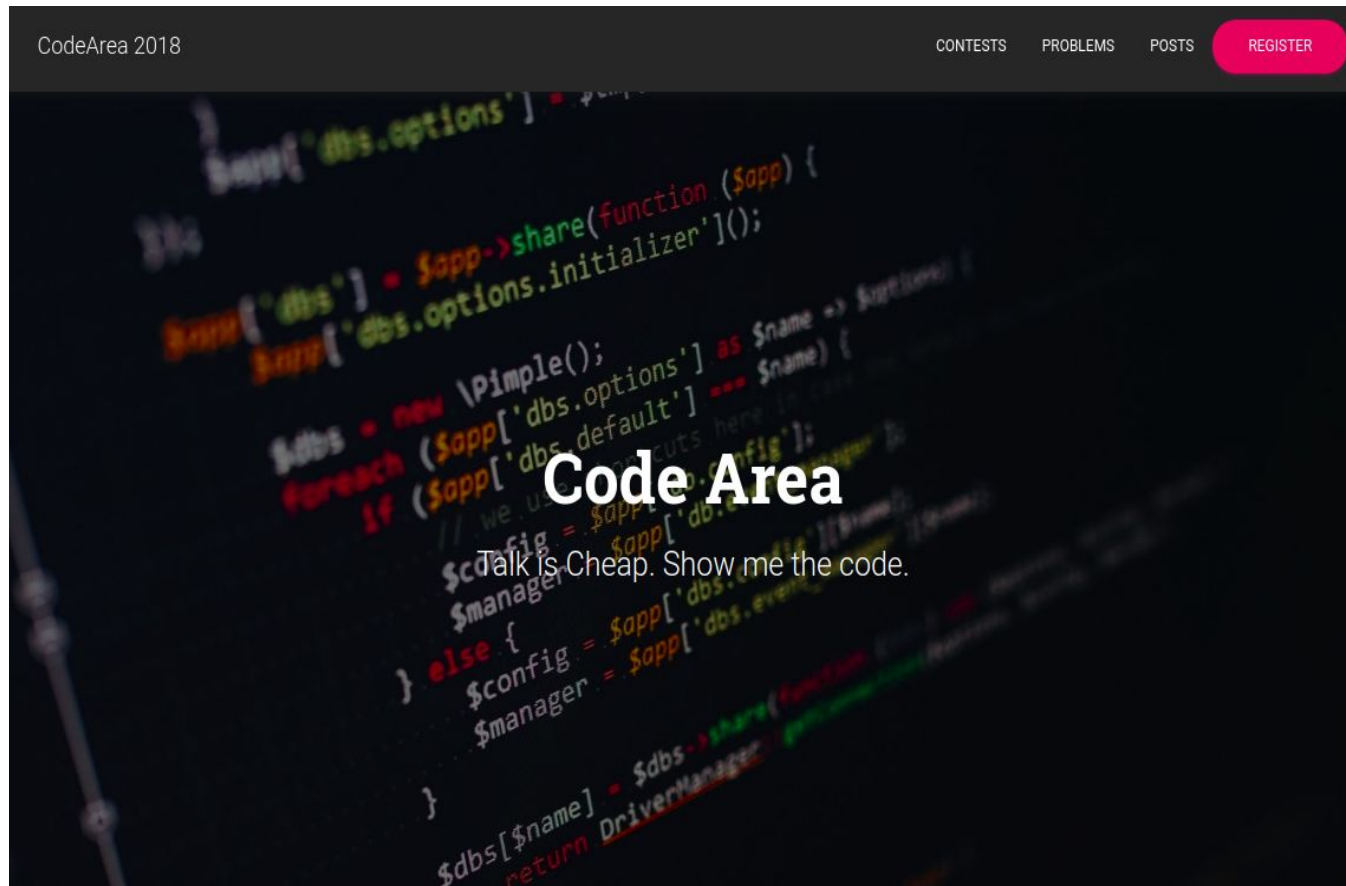
Chirag Khurana

(2016CSB1037)

Arunaksh Talukdar

(2016CSB1032)

[Project Link](#)



Guidance: [Dr. Mukesh Saini](#)