TRITRON ONLINE JUDGE

BY

Syed Mohammad Fahim Abrar
ID: 161-15-903
AND

Md Razibul Hasan Mithu
ID: 161-15-882
AND

Mohammad Shakil Mahmud
ID: 161-15-884

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised by

PROF. DR. SYED AKHTER HOSSAIN
Head
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2019

APPROVAL

This project titled "**Tritron Online Judge**", submitted by Syed Mohammad Fahim Abrar,Id:161-15-903, Md. Razibul Hasan Mithu,Id:161-15-882 and Mohammad Shakil Mahmud, Id:161-15-884 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on December, 2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. S M Aminul Haque **Internal Examiner**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Saif Mahmud Parvez **Internal Examiner**
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Prof. Dr. Syed Akhter Hossain, Head, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

____
Prof. Dr. Syed Akhter Hossain
Head
Department of CSE
Daffodil International University

Submitted by:

____
Syed Mohammad Fahim Abrar
ID: 161-15-903
Department of CSE
Daffodil International University

____
Md Razibul Hasan Mithu
ID: 161-15-882
Department of CSE
Daffodil International University

____
Mohammad Shakil Mahmud
ID: 161-15-884
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratitude to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Prof. Dr Syed Akhter Hossain, Head,** Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of web development to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Allah and also thankful to Mr Jalal Uddin, CEO**,** Devskill.com, for his kind help to give us a proper understanding of Asp.net MVC 5 to finish our project. Thanks to Zulkernine Mahmud, Software Engineer, Google to help us through proper guidance to understand multithreaded programming and Object-Oriented programming. We are also really grateful to Dr S M Aminul Hauque to give us the opportunity to host 3 Take off Programming contest and some Lab finals to prove “Tritron Online Judge” a robust system. Thanks to stack overflow community to help us by giving answers to our specific problems.

Special thanks to MoshFegh Hamedani to keep us enlighten about clean code, clean architecture and Object-Oriented Design patterns to keep the system scalable and maintainable through his online courses in pluralsight.

We would like to thank our entire programming community of Daffodil International University who contributed to testing Tritron Online Judge by attending the programming contests. Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Tritron Online Judge is an automated code judging solution. It can automate the process of checking individuals code against a set of input-output with the generated output while executing the code. Without any doubt, it will add a huge advantage in evaluating student's code in academic purpose. The automatic environment will save the time of professors on the other hand students will able to evaluate themselves immediately from the verdict of the automated judge. This will increase the algorithmic skills of students. Professor will introduce to problem settings in a professional platform, as a result, a talent problem setter community will grow. The automated judge will very cost and time-efficient for both academic and industrial purpose. The industry will be able to pre-evaluate the candidates by taking an online test of them. The endless opportunity of Tritron Online Judge makes it a great ever project.

TABLE OF CONTENTS

CONTENTS **PAGE**

CHAPTER

LIST OF FIGURES

LIST OF TABLES

TABLES **PAGE**

CHAPTER 1
INTRODUCTION

1.1 Introduction

Online Judge is a virtual automatic judge where some mathematical or algorithmic problem set will be given by a judging panel. And an individual can solve and verify their problem by submitting the solution code in some common programming language such as cplusplus, etc.and The verify process will be based on an input-output model, where some output will be generated against some input and users code will be compiled and injected by the judge's input. If the output for the injected input is similar, then it will be an accepted solution and otherwise, it will be marked with some other verdicts. Based on these verdicts online judge can give a time based marking to the participant students and thus an online programming contest can be held.

1.2 Motivation

The typical issues of manual judging of lab performance of students, for example, in traditional way lab instructors have to collect code from all computers from lab manually in storage devices ,then have to go through all the students code manually compiling them and giving input in a console and observing output of the code and judge them if the output was correct. Collecting code itself is a lengthy process then compiling one by one and judge them manually is a painful and brainstorming time killing job.

Another factor is that in this way students are bound to submit one code script for each question only. And there was no opportunity to know if their code was right or wrong in real-time. They have only the opportunity to know the final grade when the final result is published. There was no way to practice the questions later after lab evaluation finished. But solving after the exam is the most important factor to sharpen algorithmic knowledge and correction of their common mistakes.

In another way, in the traditional system, there is no advantage of doing some code rightly before other participants of the evaluation exam. But evaluation should not be the same for two students who submitted there code rightly in a time difference.

Rejudging is another big issue under the hood. Because if some instructor found that there was something dissimilarity with his questions and input-output checker data, even if he makes correction of his question again then it will be a painstaking process to rejudge the code.

So an automatic judge can take care of these issues easily and add more undeniable and powerful features.

1.3 Objectives

Our intention is to make a modern, automatic and user-friendly judging system where students will be assured if their code scripts were correct enough. It is a fairly judging process without thinking about code collecting, compiling and running by the instructor. Our judging platform will be time and cost-efficient and need less manpower to conduct any kinds of judgment at any time. Our key objectives are given below.

- Paperless exam for lab evaluation of programming and algorithmic courses.
- Time-efficient lab evaluation system.
- Realtime verdict for students code.
- Realtime rank list based on a timely fashion.
- Unlimited submission for single question script.
- Opportunity to observe previously submitted code.
- Practice offline problem-solving.
- Introducing students to programming content and online judges.
- Reduce manpower for arranging programming lab evaluation.
- Reducing human error in judging answer scripts.
- The easy rejudging system if there is a question error.

1.4 Expected Outcome

Cut down lab evaluation preparation time, logistical expenses and error-free judging platform.

1.4.1 Simple Question Setting

There will be a wysiwyg[1] editor to produce a document format of an algorithmic question, which will contain question description, input-output format, constraints of the input-output test data and lastly the sample of input-output set. There will be a file upload option to set the hidden input-output test case by which the students' code will be evaluated.

1.4.2 Varied Information Representation

Printed materials can not just suit various sorts of media. With paperless evaluation arrangements, there are huge opportunities. Non-text or graphic information, for example, audio and video can be effectively incorporated into question description. As it is very human-friendly to understand a pictorial or graphic format than the paper description it will be very handy and efficient to describe the algorithmic challenge to students.

1.4.3 Efficiency Gains

As there is no code collection portion from desk to desk of the lab, so it will be very time efficient. There is no chance to make a human error while testing input-output with judges output. The real-time verdict will help students to instantly know about their code error. And they can submit unlimited solutions for each question as the automatic judger will judge their code in a moment of eyewink.

1.4.4 Automatic Ranking

In the old school way, there was no tracking option that which student submit the accepted code in which moment. So there was no option to compare with the equally solved participant. But our system keeps track with time stamp with code submission. So we can generate a ranking table with how many tries student solved the problem and who did it fast.

1.4.5 Dynamic Rejudging

All the submissions are saved in the database with a timestamp. So rejudging the code and update rank table is just a matter of a little time. If the instructor do some mistake with making the question set or input-output set, in any time they can correct it and rejudge.

1.4.6 Cost Efficiency

With ending the lab session the final rank will publish immediately, so there is no need of manpower and time to evaluate it. Even student can also observe it with after ending the session. So it is very cost-effective as there is no need for extra examiner with an increment of the students.

1.5 Report Layout

We designed this “Tritron Online Judge” project report on the basis of six layouts. All of them are discussed below.

1.5.1 First Layout

Introduction, Motivation, Objectives and expected outcomes all these four themes are discussed in this layout.

1.5.2 Second Layout

This design explores the context of the web application, comparative studies, the nature of the issues, challenges of all these five topics.

1.5.3 Third Layout

This layout discusses business processing model, requirements, case modelling, logical data model and description of all these five topics.

1.5.4 Fourth Layout

In this format, Front & Back End development, interface and UX design, implementation specifications are addressed.

1.5.5 Fifth Layout

Implementations of Database, Front-end interactions, test output and reports all these five themes are discussed in this layout.

1.5.6 Sixth Layout

Conclusion, Discussion and scope for future work are discussed in this layout.

CHAPTER 2
BACKGROUND

2.1 Introduction

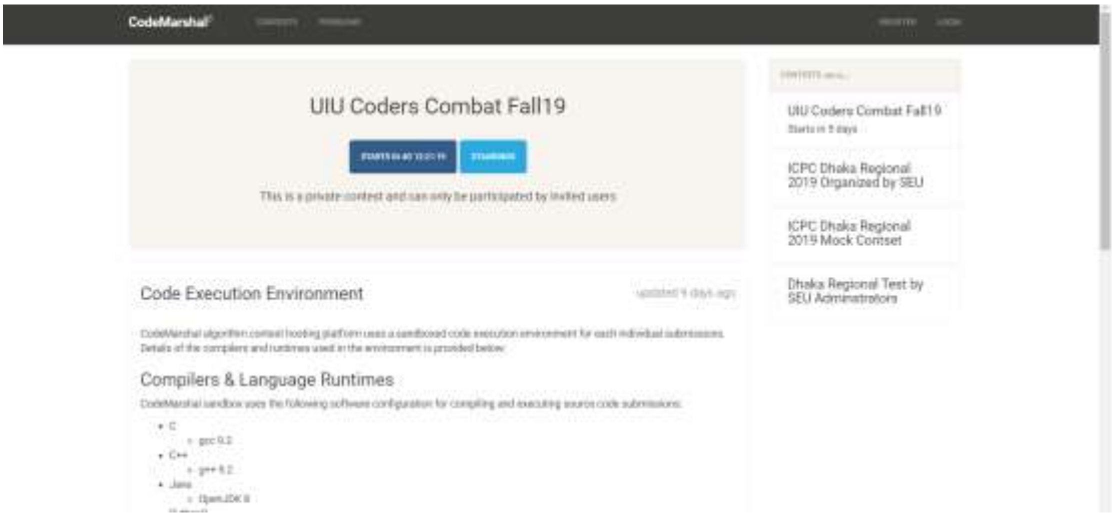
In this digital era, most of the IT/CS departments of numerous universities still go with conventional manual lab evaluation exam with manually judging students code in a compiler after collecting them from desk to desk. The idea of evaluating lab code with an automated judge did not pop out of anywhere suddenly. sports programming is known to the world from decades ago. But applying sports programming like environment in grade school’s lab evaluation will add huge advantage in growing student’s algorithmic skill and also help professors to evaluate his student easily.

Our online judge application can be used as the sports programming platform, algorithmic skill development platform, lab evaluation platform and many more.

2.2 Related Works

2.2.1 CodeMarshal

Codemarshal[1] is a contest management system widely used in programming contest like ACM-ICPC or IOI in Bangladesh. The software was built for non-profit purpose. But private contest like taking a lab final of a university is rarely allowed. All kinds of admin interaction are only limited by the site owner.


CodeMarshal
CONTESTS
STANDINGS
UIU Coders Combat Fall19
Starts in 5 days
ICPC Dhaka Regional 2019 Organized by SEU
ICPC Dhaka Regional 2019 Mock Contset
Dhaka Regional Test by SEU Administrators

UIU Coders Combat Fall19

This is a private contest and can only be participated by invited users.

Code Execution Environment

updated 9 days ago

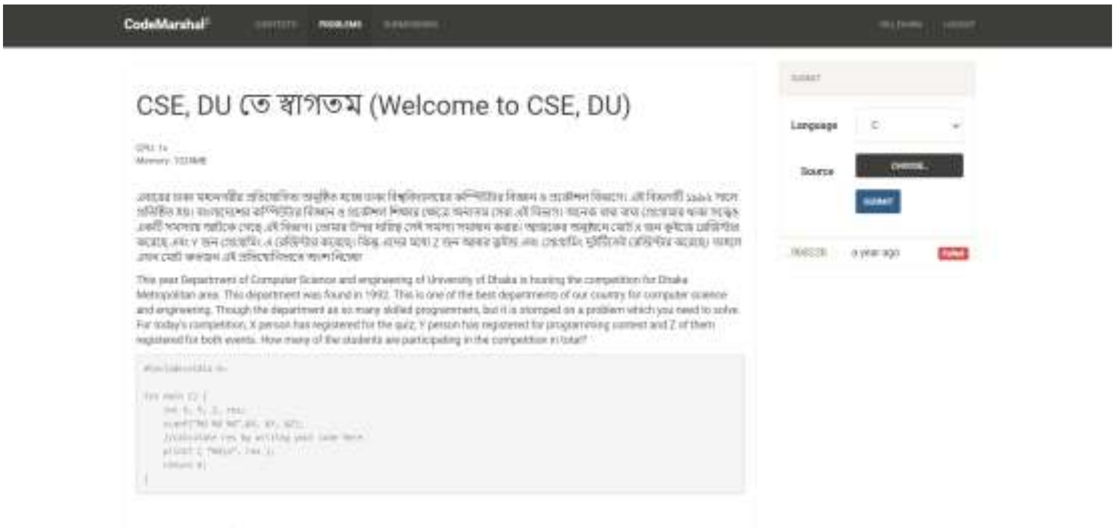
CodeMarshal algorithm contest hosting platform uses a sandboxed code execution environment for each individual submissions. Details of the compilers and runtimes used in the environment is provided below.

Compilers & Language Runtimes

CodeMarshal sandbox uses the following software configuration for compiling and executing source code submissions.

- C
  - gcc 9.2
- C++
  - g++ 9.2
- Java
  - OpenJDK 8

Figure 2.1: CodeMarshal Dashboard



CSE, DU তে স্বাগতম (Welcome to CSE, DU)

CPU: 1s
Memory: 512MB

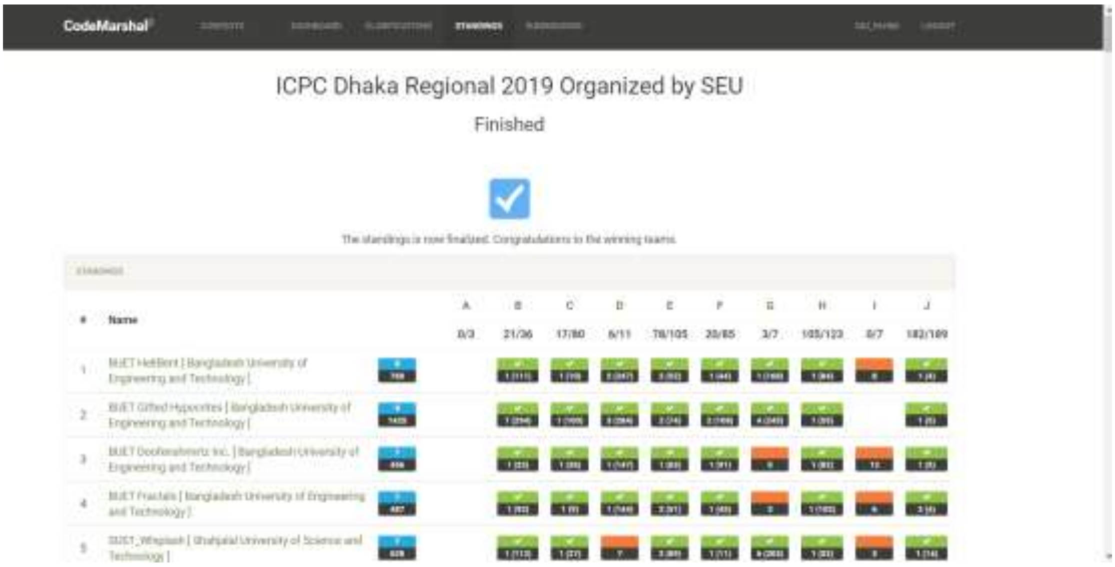
[illegible]

This year Department of Computer Science and engineering of University of Dhaka is hosting the competition for Dhaka Metropolitan area. This department was found in 1992. This is one of the best departments of our country for computer science and engineering. Though the department as so many skilled programmers, but it is stomped on a problem which you need to solve. For today's competition, X person has registered for the quiz, Y person has registered for programming contest and Z of them registered for both events. How many of the students are participating in the competition in total?

```
#include<stdio.h>

int main () {
    int X, Y, Z, res;
    scanf("%d %d %d",&X, &Y, &Z);
    //calculate res by writing your code here
    printf ( "%d\n", res );
    return 0;
}
```

Figure 2.2: CodeMarshal Problem Page



ICPC Dhaka Regional 2019 Organized by SEU

Finished

The standings is now finalized. Congratulations to the winning teams.

STANDINGS

#	Name		A 0/3	B 21/36	C 17/80	D 6/11	E 78/105	F 20/85	G 3/7	H 105/123	I 0/7	J 182/189
1	BUET Hellbent [Bangladesh University of Engineering and Technology]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	8	[illegible]
2	BUET Gifted Hypocrites [Bangladesh University of Engineering and Technology]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]
3	BUET [illegible] Inc. [Bangladesh University of Engineering and Technology]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	9	[illegible]	10	[illegible]
4	BUET Fractals [Bangladesh University of Engineering and Technology]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3	[illegible]	6	[illegible]
5	SUST_Whiplash [Shahjalal University of Science and Technology]	[illegible]		[illegible]	[illegible]	7	[illegible]	[illegible]	[illegible]	[illegible]	8	[illegible]

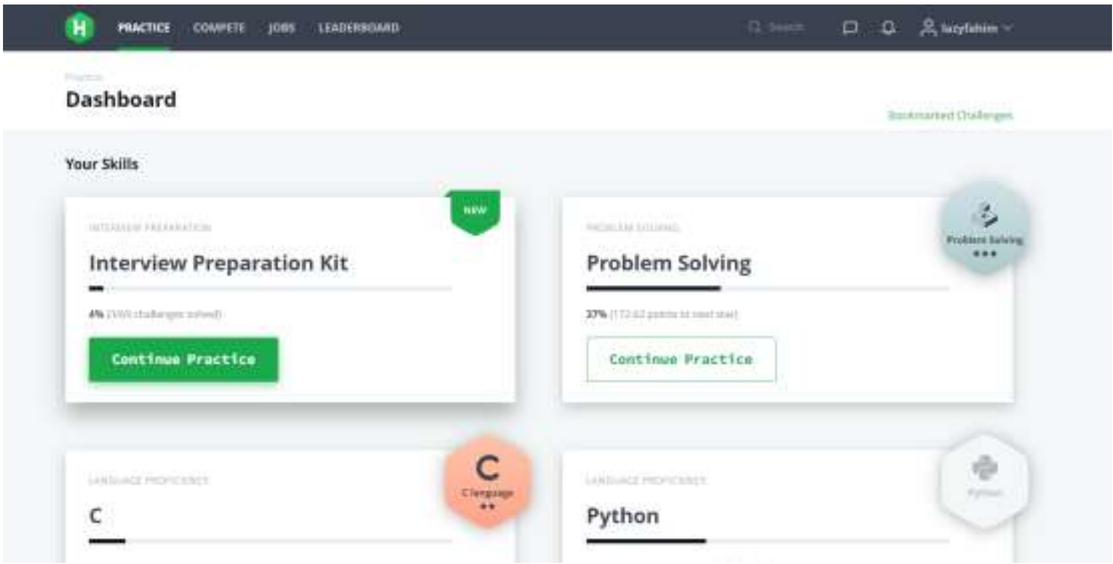
Figure 2.3: CodeMarshal Standing

2.2.2 HackerRank

HackerRank[2] is an Indian algorithmic problem practice website. It has an algorithmic problem practising arena and public contest arranging feature. But no private contest is allowed.

It can host only contest with IOI style leaderboard.

Related Image:



Dashboard

Your Skills

INTERVIEW PREPARATION

Interview Preparation Kit

4% (3/69 challenges solved)

Continue Practice

PROBLEM SOLVING

Problem Solving

37% (172.62 points to next star)

Continue Practice

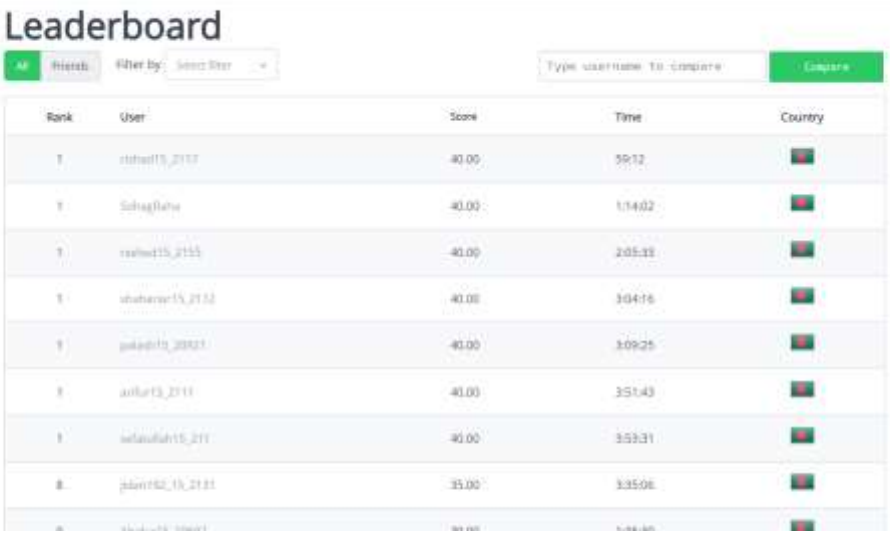
LANGUAGE PROFICIENCY

C

LANGUAGE PROFICIENCY

Python

Figure 2.4: Hackerrank Dashboard



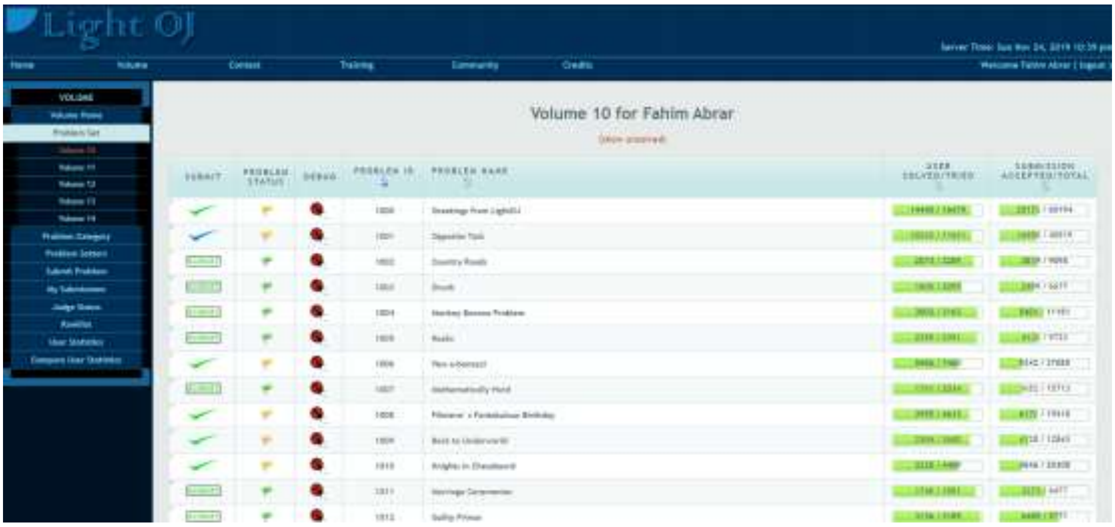
Leaderboard

Rank	User	Score	Time	Country
1	rishad15_2117	40.00	59:12	🇧🇩
1	SohagRana	40.00	1:14:02	🇧🇩
1	rashed15_2155	40.00	2:05:33	🇧🇩
1	shahariar15_2132	40.00	3:04:16	🇧🇩
1	prakash15_20921	40.00	3:09:25	🇧🇩
1	arifur15_2111	40.00	3:51:43	🇧🇩
1	sefatullah15_211	40.00	3:53:31	🇧🇩
8	jidan162_15_2131	35.00	3:35:06	🇧🇩
[illegible]	[illegible]	[illegible]	[illegible]	🇧🇩

Figure 2.5: Hackerank LeaderBoard

2.2.3 LightOJ

Lightoj[3] is a Bangladeshi Online judge with numerous featureful problem that are useful to learn problem-solving. Having solved count in this OJ is an advantage on the Bangladeshi job market. Though it has not enough powerfull contest service as others.

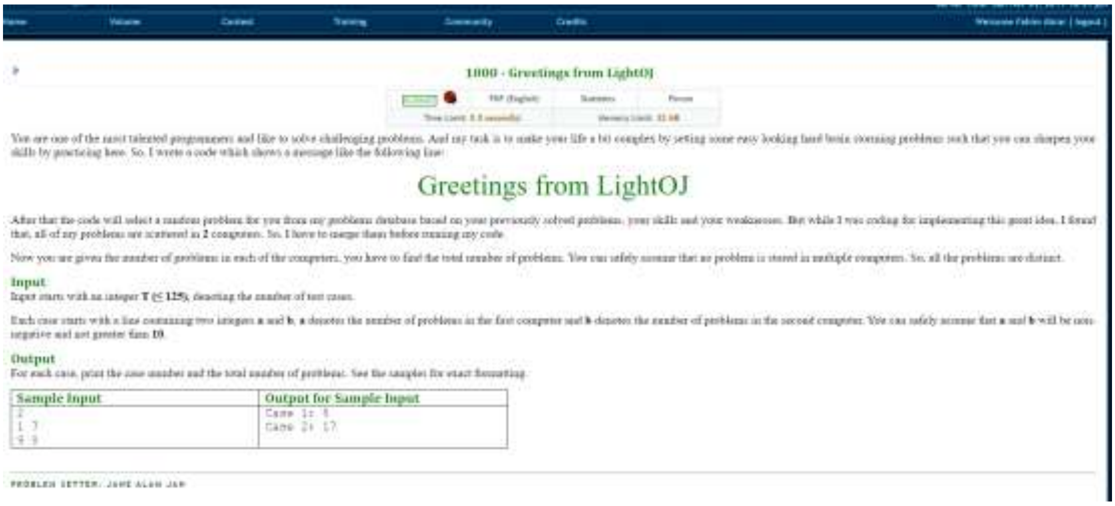


Volume 10 for Fahim Abrar

(show unsolved)

SUBMIT	PROBLEM STATUS	DEBUG	PROBLEM ID	PROBLEM NAME	USER SOLVED/TRIED	SUBMISSION ACCEPTED/TOTAL
✓	●	●	1000	Greetings from LightOJ	[illegible]	[illegible]
✓	●	●	1001	Opposite Task	[illegible]	[illegible]
Submit	●	●	1002	Country Roads	[illegible]	[illegible]
Submit	●	●	1003	Drunk	[illegible]	[illegible]
Submit	●	●	1004	Monkey Banana Problem	[illegible]	[illegible]
Submit	●	●	1005	Rooks	[illegible]	[illegible]
✓	●	●	1006	Hex-a-bonacci	[illegible]	[illegible]
Submit	●	●	1007	Mathematically Hard	[illegible]	[illegible]
✓	●	●	1008	Fibsieve`s Fantabulous Birthday	[illegible]	[illegible]
✓	●	●	1009	Back to Underworld	[illegible]	[illegible]
✓	●	●	1010	Knights in Chessboard	[illegible]	[illegible]
Submit	●	●	1011	Marriage Ceremonies	[illegible]	[illegible]
Submit	●	●	1012	Guilty Prince	[illegible]	[illegible]

Figure 2.6: Lightoj Dashboard



1000 - Greetings from LightOJ

Time Limit: 0.5 second(s) Memory Limit: 32 MB

You are one of the most talented programmers and like to solve challenging problems. And my task is to make your life a bit complex by setting some easy looking hard brain storming problems such that you can sharpen your skills by practicing here. So, I wrote a code which shows a message like the following line:

Greetings from LightOJ

After that the code will select a random problem for you from my problems database based on your previously solved problems, your skills and your weaknesses. But while I was coding for implementing this great idea, I found that, all of my problems are scattered in 2 computers. So, I have to merge them before running my code.

Now you are given the number of problems in each of the computers, you have to find the total number of problems. You can safely assume that no problem is stored in multiple computers. So, all the problems are distinct.

Input

Input starts with an integer T (≤ 125), denoting the number of test cases.

Each case starts with a line containing two integers a and b. a denotes the number of problems in the first computer and b denotes the number of problems in the second computer. You can safely assume that a and b will be non-negative and not greater than 10.

Output

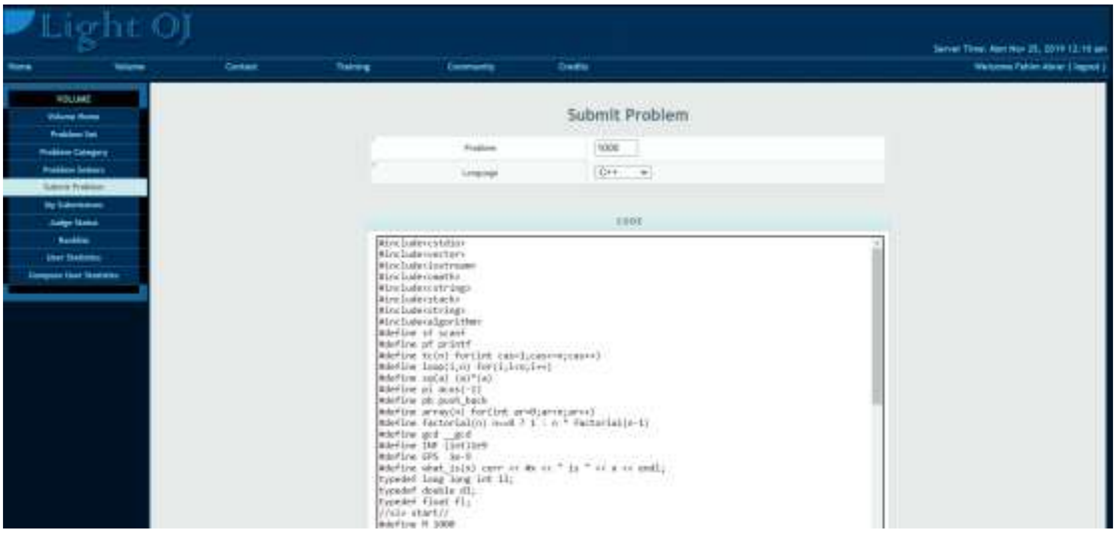
For each case, print the case number and the total number of problems. See the samples for exact formatting.

Sample Input	Output for Sample Input

2
1 7
9 8
	Case 1: 8
	Case 2: 17

PROBLEM SETTER: JANE ALAM JAN

Figure 2.7:Lightoj problem page



Submit Problem

Problem: 1000

Language: C++

CODE

```
#include<cstdio>
#include<vector>
#include<iostream>
#include<cmath>
#include<cstring>
#include<stack>
#include<string>
#include<algorithm>
#define sf scanf
#define pf printf
#define tc(n) for(int cas=1;cas<=n;cas++)
#define loop(i,n) for(i=1;i<=n;i++)
#define sq(a) (a)*(a)
#define pi acos(-1)
#define pb push_back
#define array(n) for(int a=0;a<n;a++)
#define factorial(n) n==0 ? 1 : n * factorial(n-1)
#define gcd __gcd
#define INF (int)1e9
#define EPS 1e-9
#define what_is(x) cerr << #x << " is " << x << endl;
typedef long long int ll;
typedef double dl;
typedef float fl;
//sls start//
#define M 1000
```

Figure 2.7:Lightoj Submission page

2.3 Comparative Studies

We have analyzed some of the similar online judges. They have some awesome page but not in one package. We gained ideas about how to integrate the crucial features all in one package in our online judge as an enterprise application.

Table 2.1: Comparative Studies

Web-Application	Name	Key Features	Pros & Cons
Related	CodeMarshal	Online programming contest, Offline Problem solving, IOI/ACM style ranking, Live standing with Websocket.	It covers all necessary features but only private contest can be hosted by one admin which is not featureful for individual University.
Related	HackerRank	Programming Contest, Online CodeEditor, Custom Slug, Public Contest, Offline Problem Solving Tutorial	An individual can arrange a contest but contests are always public and allowed for all, Standing is not live(User have to refresh page)
Related	Lightoj	Offline Problem Solving, Contest hosting in a limited form.	It's not for all kind programming contest, Huge database for

			algorithmic problem
Our System	Tritron Online Judge	Offline problem solving, Online Contest, Onsite Contest, Live Standing With WebSocket, On-Demand User Id generation on contest floor, Code SandBoxing	This system can arrange any kind of online and onsite contest,User doesn't have to refresh to see live updates.

2.4 Scope of the Problem

Tritron Online Judge is mainly focused on hosting programming contest, taking lab evaluation, Offline algorithmic problem solving, while making this app we have to face many troubles and problem given below.

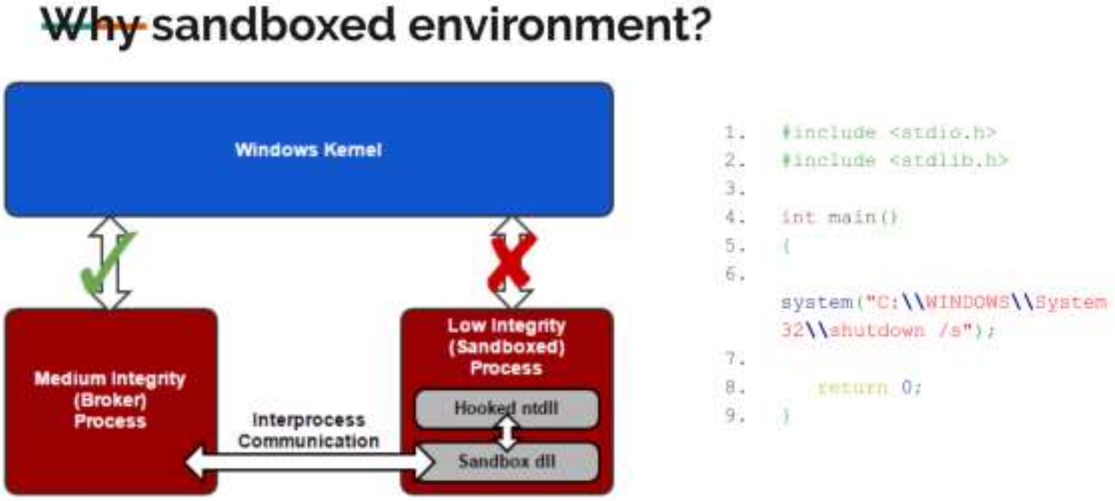
- Code Sandboxing for malicious code
- Multi-Threading
- Visibility problem of the multithreaded judger.
- Flag continuation in the compiler.

2.5 Challenges

We analyze before building this application to find out the challenges.

2.5.1 Code Sandboxing for malicious code

Running user's code itself malicious for any production environment. Because any administrative access by users code they can delete files or folder or modify the result of the contest by a system call. So we have to ensure that users code cannot make any system call and cannot do modification outside of its own sandboxed environment.



~~Why~~ sandboxed environment?

Windows Kernel
Medium Integrity (Broker) Process
Low Integrity (Sandboxed) Process
Hooked ntdll
Sandbox dll
Interprocess Communication

```
#include <stdio.h>
#include <stdlib.h>

int main()
{

system("C:\\WINDOWS\\System
32\\shutdown /s");

   return 0;
}
```

Figure 2.9:Sandboxing

2.5.2 Multithreading

We had to implement a multithreaded worker process to compile, execute and judge contestants code. In this way, we were able to process the verdict in a non-blocking way.

2.5.3 Visibility Problem

In a multithreaded environment, one thread doesn't know about other threads. So we had to generate a synchronized hash table to track that which thread was handling which submission. But when a thread was locking one entity from database in the meantime if the judge or admin modified any input-output data, the data changed was not effected in the thread. So, at last, we used the volatile variable to solve this problem.

2.5.4 Flag continuation in the compiler

We had to handle various compiler flag as usual used in programming contest such as #ifndef ONLINE_JUDGE, #define pie, faster I/O etc.we passed them as command-line arguments in the compiler.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 Business Process Modeling

Business procedure modelling speaks to the business procedure or work process as a method for recognizing potential enhancements. The business process modelling is given below:

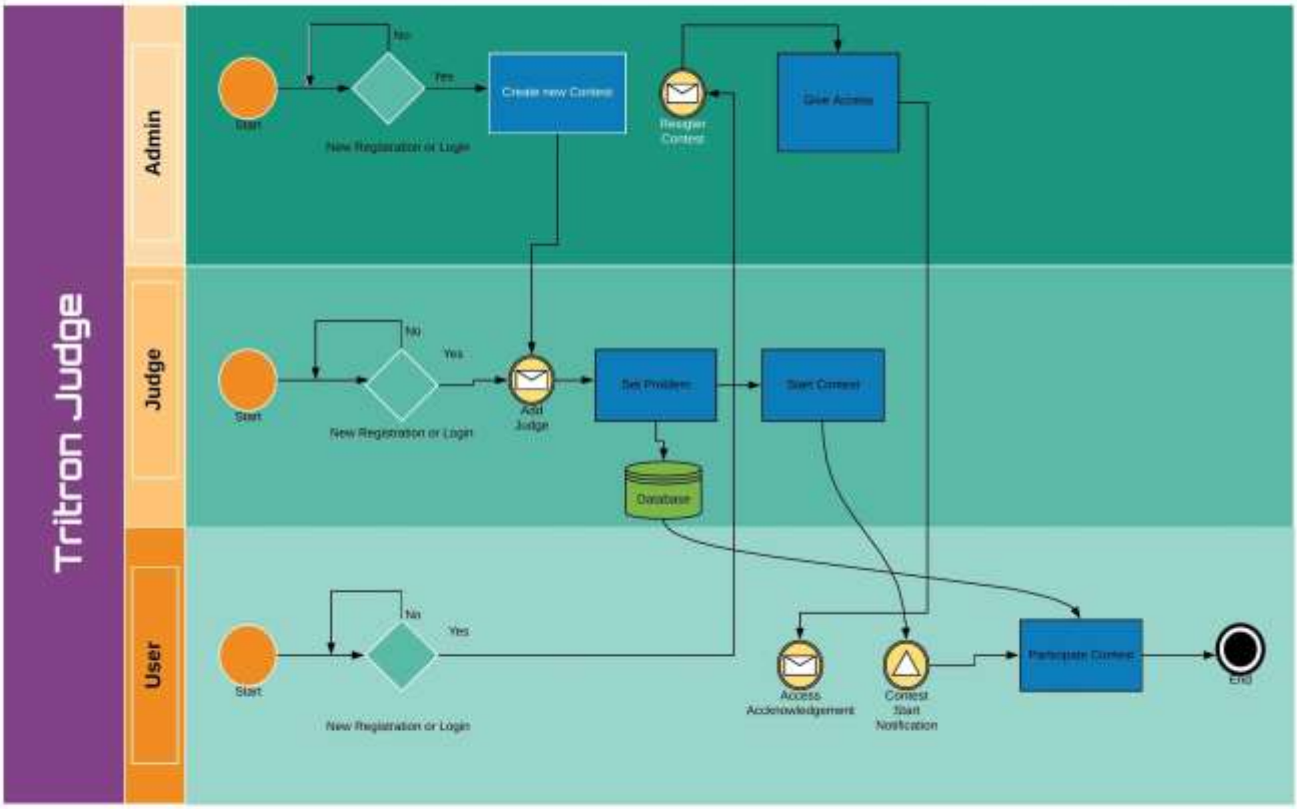


Figure 3.1: Business Process Modeling

3.2 Requirement Collection and Analysis

3.2.1 Pre-Contest Process

- Create an algorithmic problem
- Create Programming contest
- Give access to judge in the programming contest
- Add custom generate user in the onsite contest
- Register for the contest

3.2.2 During Contest Process

- Register for the contest.
- Solve algorithmic problems.
- Submit code.
- Generate standing.

- See users code
- See own code
- Edit Problem statement

3.2.3 Post-Contest Processes

- Offline Problem solve.
- Review Users code.
- See Standing.
- Generate contest report.

3.3 Use Case Modeling and Description

By drawing this diagram, we showed the activities of a client. We implemented use case diagram because it can find out what are the framework requirements are and utilize them.

3.3.1 Use Case of Admin

The use case diagram of an admin is given below:

Table 3.1: Use Case of Admin

Use case name	Use Case of Admin
Actor	Admin, Judge
Description	Admin will handle all the work procedures.
Pre-conditions	Must have an ID and password.
Standard flow	1. Have to log in first. 2. Can Create algorithmic problems. 3. Can arrange a contest. 4. Can add judge for the contest. 5. Review users code. 6. Can create custom bulk users for the onsite contest.

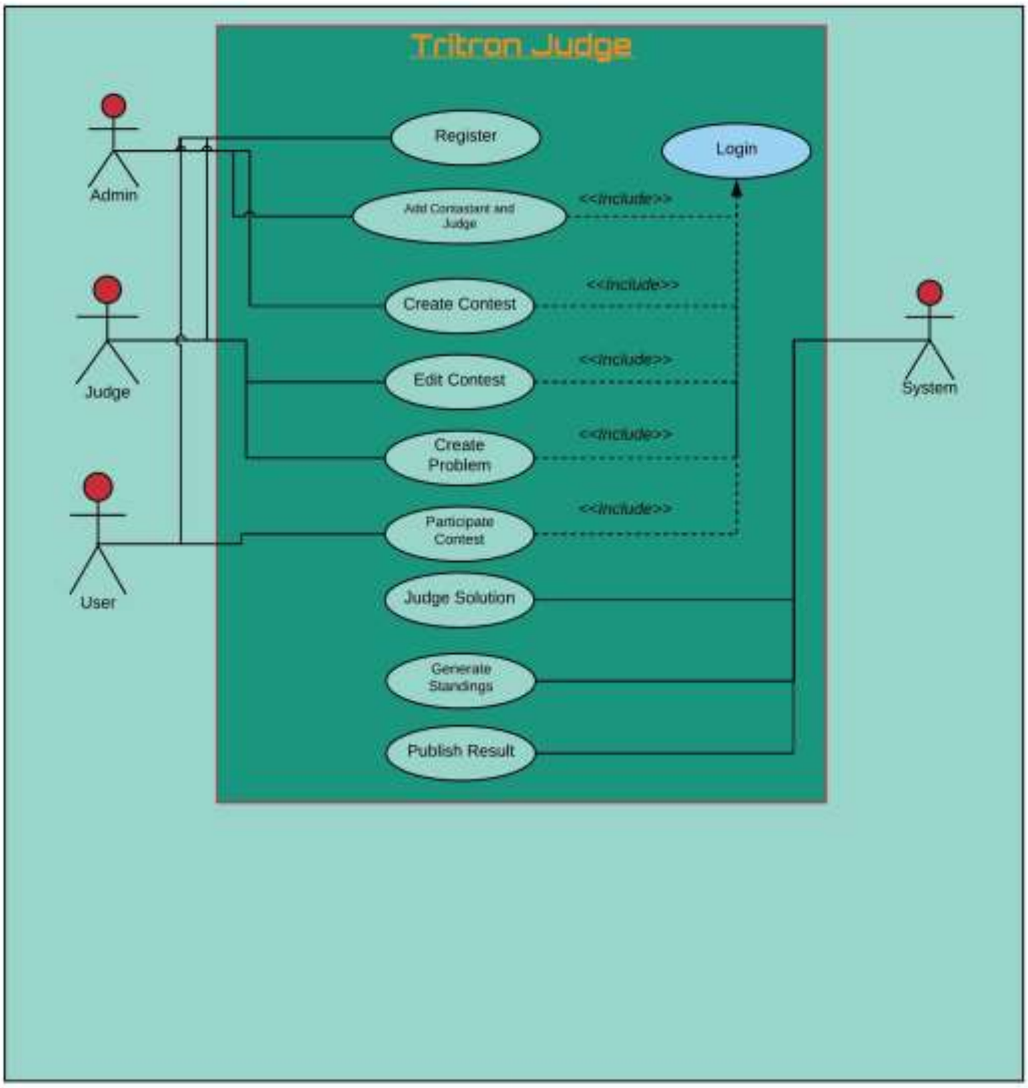


Figure 3.2: Use case of Triton Online Judge

3.3.2 Use Case of User

The use case diagram of a user is given below:

Table 3. 2: Use Case of User

Use case name	Use Case of User
Actor	User
Description	Access for the registered contest and can up solve problems after the contest.

Pre-conditions	Must have an ID and password.
Standard flow	1. Have to log in first. 2. Can view offline problems. 3. User can register for an online contest. 4. Only selected or generated user will have access to private/onsite contest. 5. Can submit solution in programming language.

3.4 Logical Data Model

Logical data model portrays the information however much detail as could reasonably be expected. The Entity-Relationship diagram and the Logical Schema are the logical data model which includes attributes, entity, tables and relationships.

3.4.1 Entity Relationship Diagram

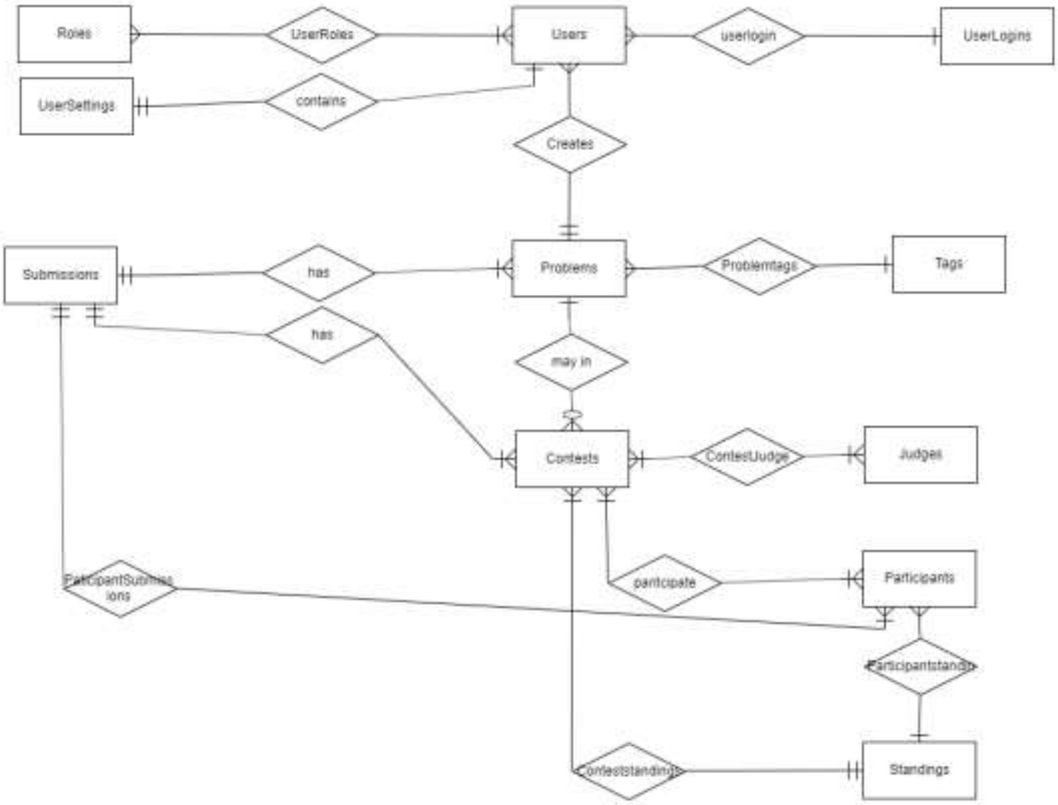
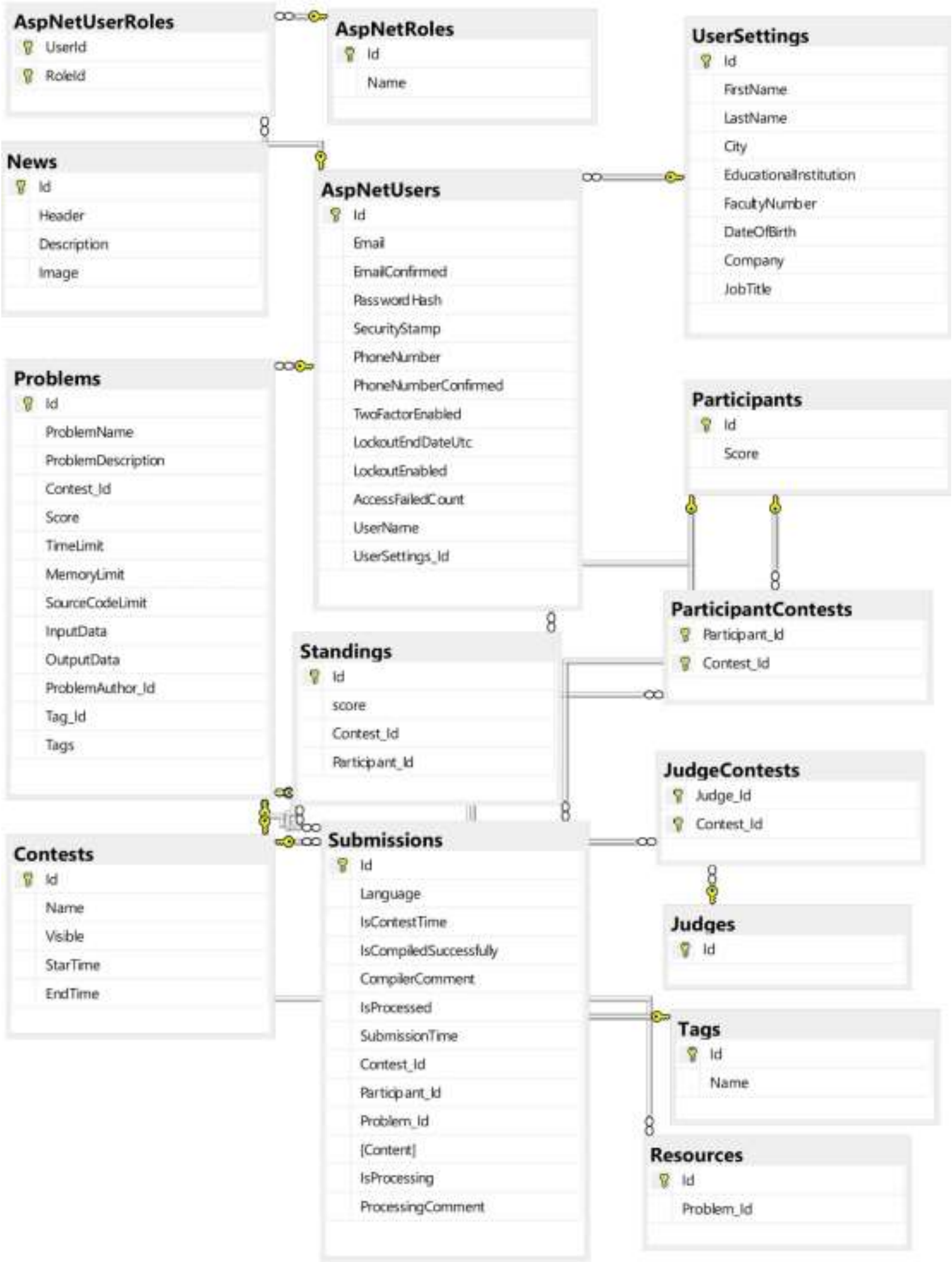


Figure 3.3: Entity Relationship Diagram

3.4.2 Logical Schema

The Logical Schema is given below:



1-1

Figure 3.4: Logical Schema

3.4.1 Data Flow Diagram

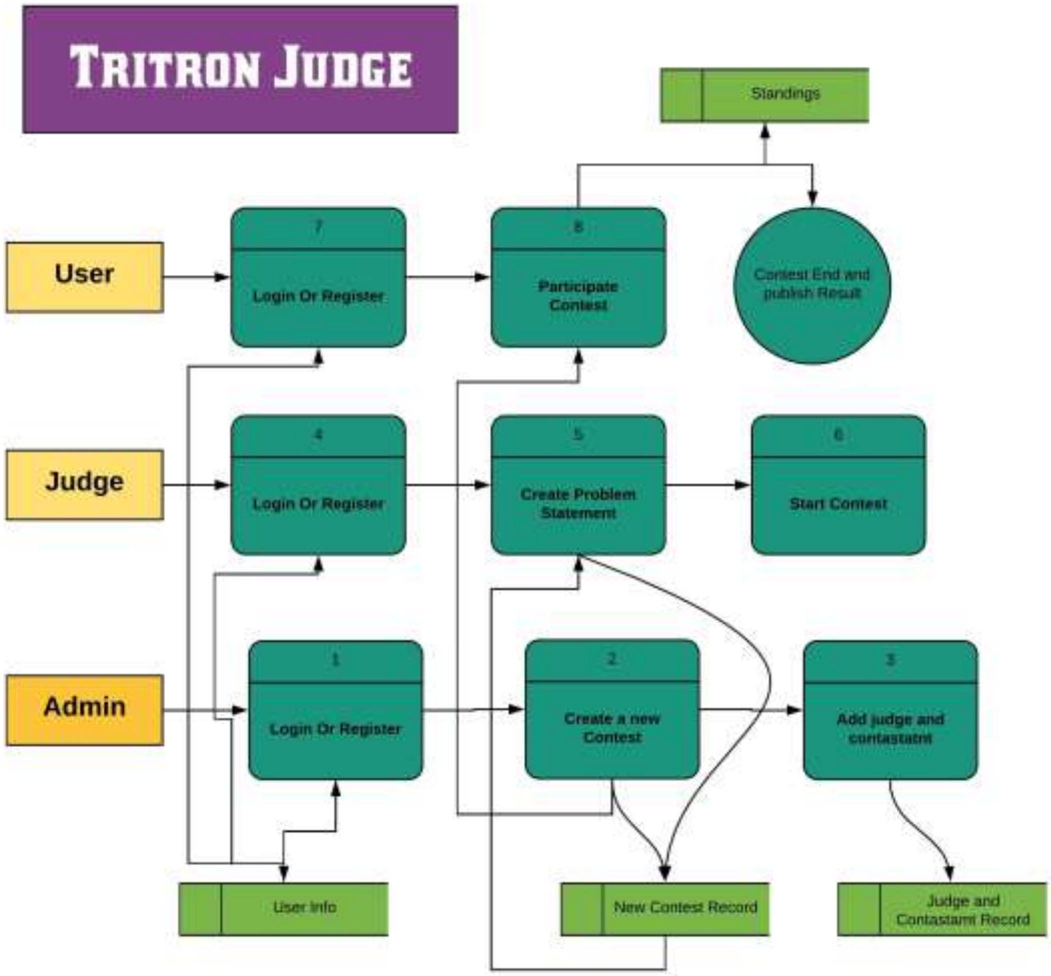


Figure 3.5: Data Flow Diagram

3.5 Design Requirements

Initially, we were fixed the following requirements to implement in our project.

1. User-friendly site.
2. Easy way to create a contest.
3. Easy way to create problems.
4. Easy way to participate in the contest.
5. Pleasant dashboard.
6. Show code verdict and live standing.

CHAPTER 4
DESIGN SPECIFICATION

4.1 Front-end Design

The basic requirement of "Tritron Online Judge" front end design contains webpages which are designed initially by sketching and we implemented them later where we used the bootstrap 4 framework and then developed implementing jquery framework.

4.1.1 HTML

To implement the user interface of our app we wrote web pages using HTML to build the basic structure of our app as HTML is the skeleton of any web app. We optimized the HTML codes in a way that is completely readable to anyone.[4]

4.1.2 CSS

We implemented CSS to transform our web pages into dynamic web pages. Which described the presentation of html codes and the documents in a page including colours, layout and fonts. Which allows presentation in different types of devices.[5]

4.1.3 Bootstrap

Bootstrap is an open-source trendiest, responsive and smart front-end framework as its mobile-first, prevailing. It developed along with HTML, SaaS and JavaScript and easy to implement as such we used it in our project which made our project responsive and developed the speed of our system. The other advantages it provides us with the capability of utilizing 12-column grid and customizable.[6]

4.1.4 JavaScript

JavaScript allows creating highly responsive user interfaces which let dynamic functionality as we don't need to wait for the server to react to show another task. Besides containing better functionality and libraries it made our task easy to implement.[7]

4.2 Back-end Design

Backend development is an aptitude that powers the web. However, it does it unobtrusively, without flourish — allowing people to browse their favourite sites without thinking pretty much all the work put in by the backend developer or group.

4.2.1 Asp.net MVC5

We implemented our web app part in Asp.net MVC5. We have created our business logics and database schemas in .Net Class Library type project. The class library projects and logical schemas had abstraction with web app presentation layer. We had used ninject dependency injection framework to avoid memory leakage with class library projects. [8]

4.2.2 SQL Server and Entity Framework

We had to use the MSSQL server as backend database and entity framework is an object-relational mapping for avoiding writing raw SQL code.[9]

4.2.3 Identity Framework

We chose Microsoft's identity framework to handle User authorization, sign in, sign up, email sending, account activation and various kind of authentication-related works.

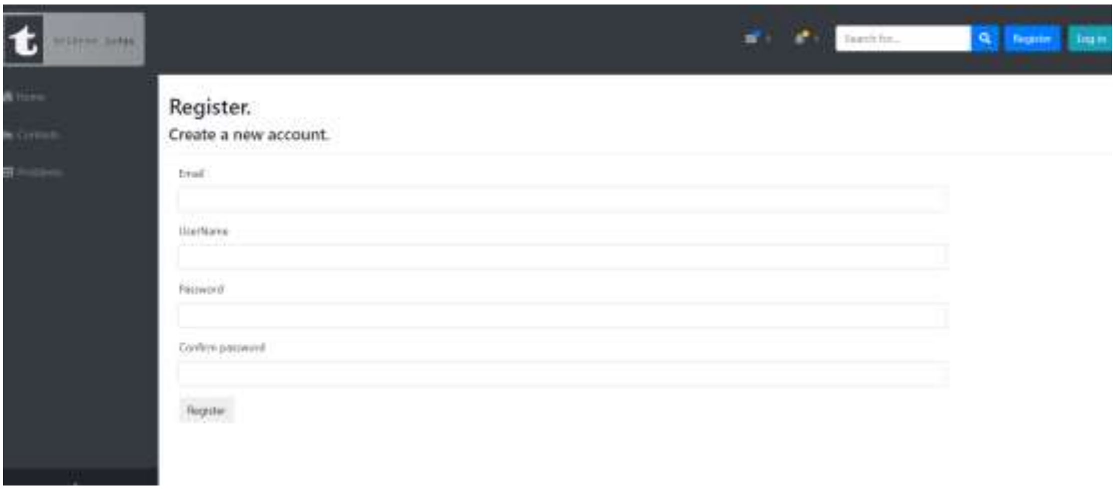
4.2.4 SignalR and SQL Dependency

As one of our prerequisites was to implement a real-time standing system user doesn't have to page reload to see the latest standing, we implemented long polling and websocket with SQL dependency and SignalR.[10]

4.3 Interaction Design and UX

User Interface (UI) design is the method of making interfaces in programming or modernized gadgets with an attention on looks or style. UI design usually refers to graphical user interfaces. It is the most significant part for any site for guaranteeing better client experience.

4.3.1 Register Page



Register.

Create a new account.

Figure 4.1: Register Page

4.3.2 Sign in Page

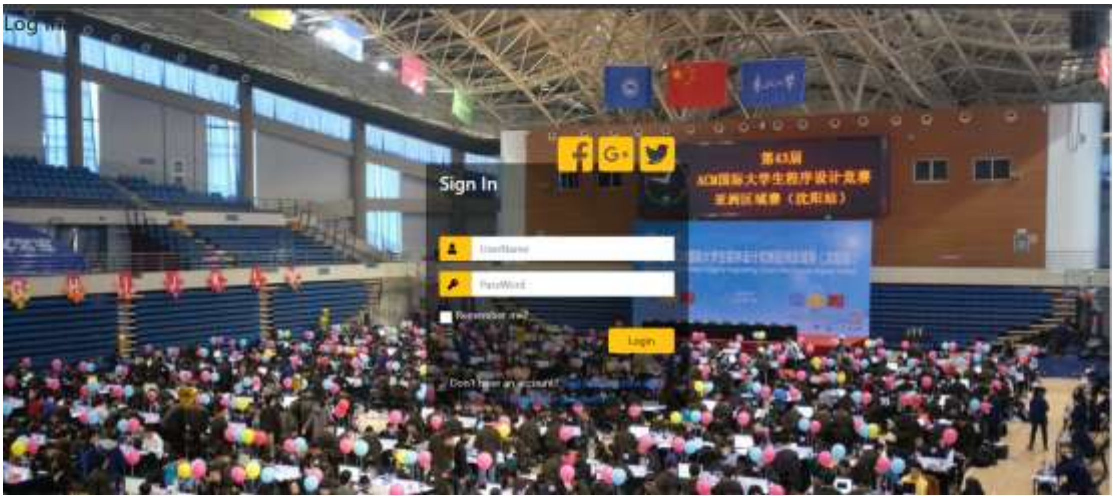
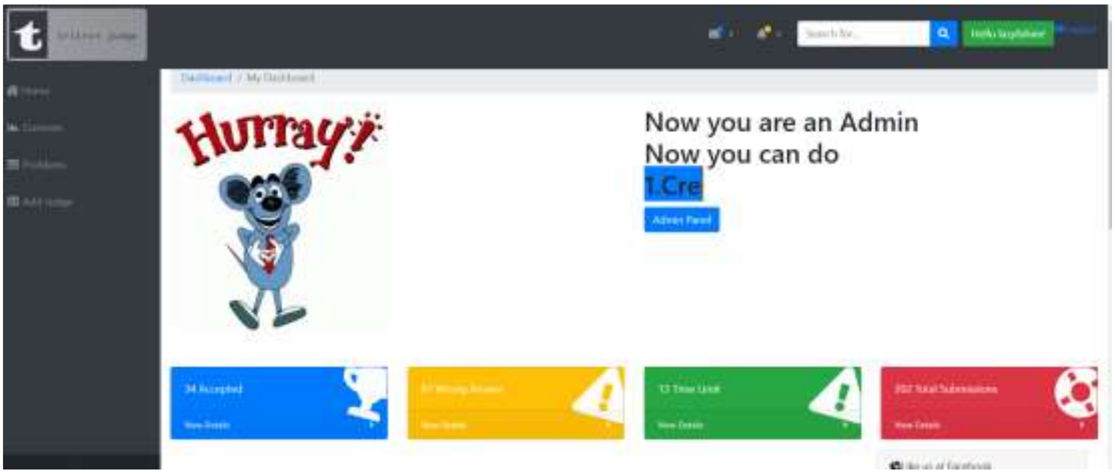


Figure 4.2: Log in Page

4.3.3 Dashboard Page

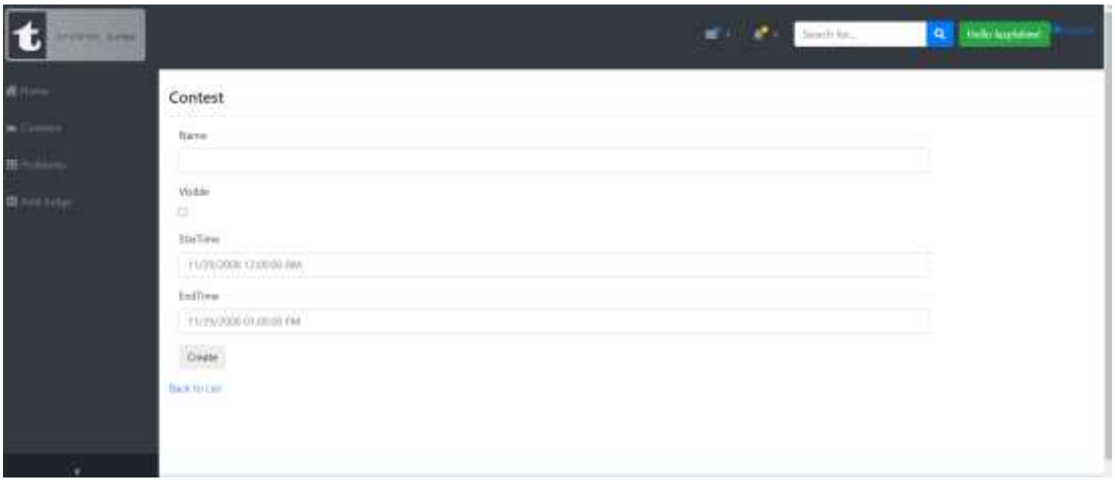


Dashboard / My Dashboard

Now you are an Admin
Now you can do
1.Cre

Figure 4.3: Dashboard Page

4.3.4 Create Meeting Page



Contest

Name

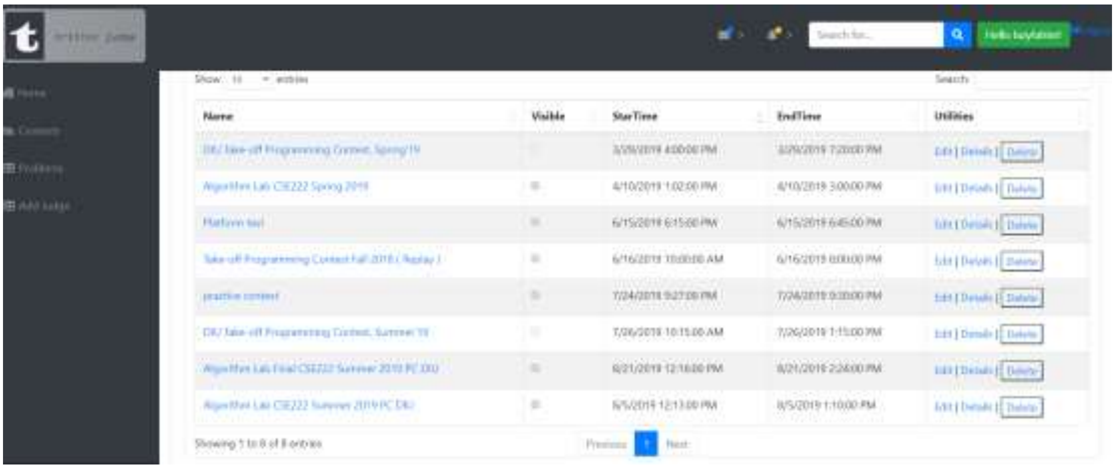
Visible
☐

StarTime
11/29/2006 12:00:00 AM

EndTime
11/29/2006 01:00:00 PM

Figure 4.4: Create Contest Page

4.3.5 All Contest Page



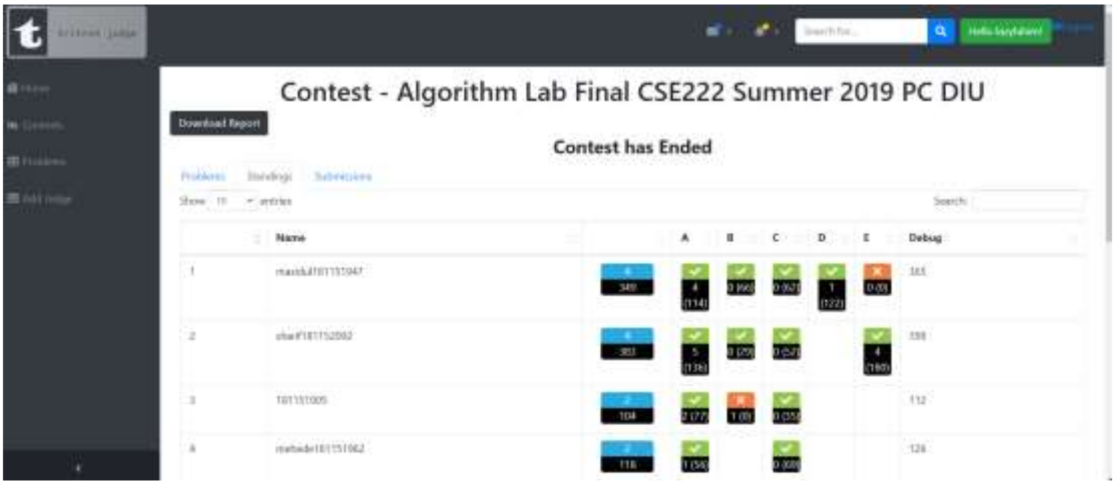
Show 10 entries

Name	Visible	StarTime	EndTime	Utilities
DIU Take-off Programming Contest, Spring'19	☐	3/29/2019 4:00:00 PM	3/29/2019 7:20:00 PM	Edit | Details | Delete
Algorithm Lab CSE222 Spring 2019	☑	4/10/2019 1:02:00 PM	4/10/2019 3:00:00 PM	Edit | Details | Delete
Platform test	☑	6/15/2019 6:15:00 PM	6/15/2019 6:45:00 PM	Edit | Details | Delete
Take-off Programming Contest Fall 2018 (Replay)	☑	6/16/2019 10:00:00 AM	6/16/2019 8:00:00 PM	Edit | Details | Delete
practice contest	☑	7/24/2019 9:27:00 PM	7/24/2019 9:30:00 PM	Edit | Details | Delete
DIU Take-off Programming Contest, Summer 19	☐	7/26/2019 10:15:00 AM	7/26/2019 1:15:00 PM	Edit | Details | Delete
Algorithm Lab Final CSE222 Summer 2019 PC DIU	☑	8/21/2019 12:16:00 PM	8/21/2019 2:24:00 PM	Edit | Details | Delete
Algorithm Lab CSE222 Summer 2019 PC DIU	☑	8/5/2019 12:13:00 PM	8/5/2019 1:10:00 PM	Edit | Details | Delete

Showing 1 to 8 of 8 entries

Figure 4.5: All Contest Page

4.3.6 Live Standing Page



Contest - Algorithm Lab Final CSE222 Summer 2019 PC DIU

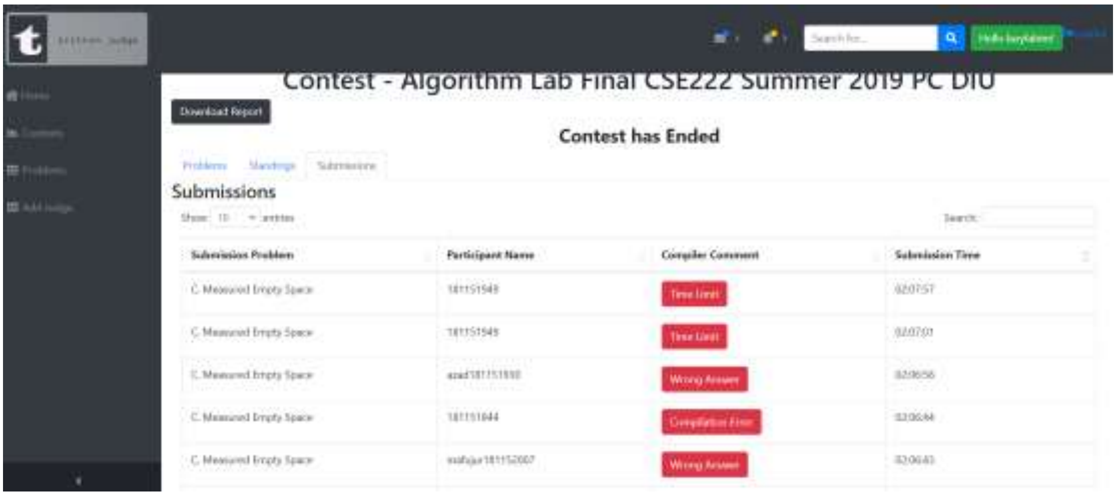
Contest has Ended

	Name		A	B	C	D	E	Debug
1	mazidul181151947	4 349	✓ 4 (114)	✓ 0 (66)	✓ 0 (62)	✓ 1 (122)	✗ 0 (0)	165
2	sharif181152082	4 383	✓ 5 (136)	✓ 0 (29)	✓ 0 (57)		✓ 4 (180)	158
3	181151005	2 104	✓ 2 (77)	✗ 1 (0)	✓ 0 (35)			112
4	mahede181151062	2 116	✓ 1 (56)		✓ 0 (60)			126

Figure 4.6:

Live Standing Page

4.3.7 Submissions in contest Page



Contest - Algorithm Lab Final CSE222 Summer 2019 PC DIU

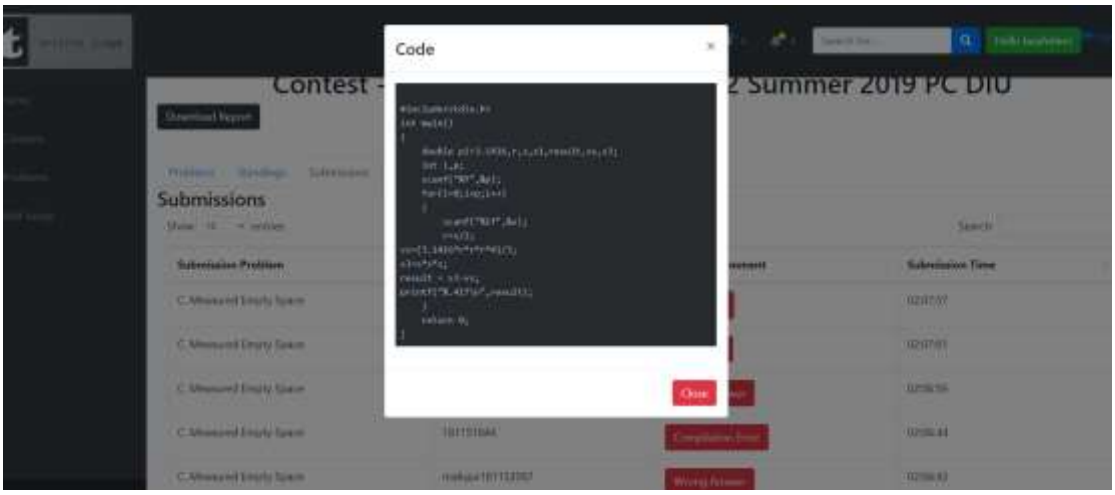
Contest has Ended

Submissions

Submission Problem	Participant Name	Compiler Comment	Submission Time
C. Measured Empty Space	181151949	Time Limit	02:07:57
C. Measured Empty Space	181151949	Time Limit	02:07:01
C. Measured Empty Space	azad181151930	Wrong Answer	02:06:56
C. Measured Empty Space	181151844	Compilation Error	02:06:44
C. Measured Empty Space	mahjur181152007	Wrong Answer	02:06:43

Figure 4.7: Submissions in contest Page

4.3.7 Submission view Modal

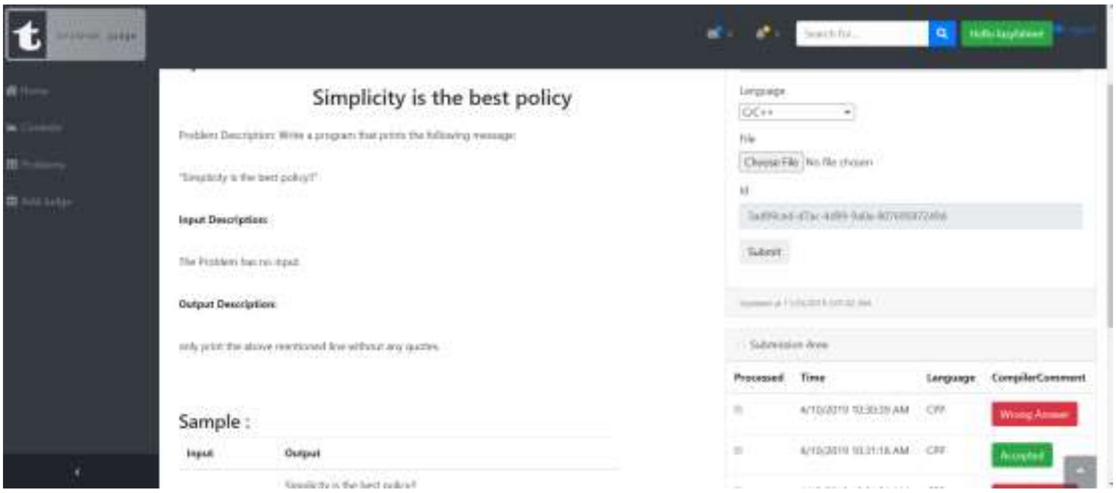


```
#include<stdio.h>
int main()
{
    double pi=3.1416,r,s,s1,result,ss,s2;
    int i,n;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%lf",&r);
        r=r/2;
ss=(3.1416*r*r*r)/2;
s1=s*s*s;
result = s1-ss;
printf("%.4lf\n",result);
    }
    return 0;
}
```

Submission Problem		Comment	Submission Time
C.Measured Empty Space			02:07:57
C.Measured Empty Space			02:07:01
C.Measured Empty Space			02:08:56
C.Measured Empty Space	181151844	Compilation Error	02:06:44
C.Measured Empty Space	[illegible]181153187	Wrong Answer	02:06:43

Figure 4.8: Submissions view in Modal

4.3.7 Problem Page



Simplicity is the best policy

Problem Description: Write a program that prints the following message:

"Simplicity is the best policy!"

Input Description:

The Problem has no input.

Output Description:

only print the above mentioned line without any quotes.

Sample :

Input	Output
	Simplicity is the best policy!

Processed	Time	Language	CompilerComment
☑	4/10/2019 10:30:39 AM	CPP	Wrong Answer
☑	4/10/2019 10:31:16 AM	CPP	Accepted

Figure 4.9: Problem View Page

4.4 Implementation Requirements

Following technologies used in my project.

1. HTML 5 and CSS
2. Bootstrap 4
3. JavaScript
4. ASP.NET MVC 5
5. Identity Framework
6. Windows service worker
7. Entity Framework
8. SignalR

The following software used to implement the application:

1. Visual Studio Enterprise 2019
2. Git
3. Azure Devops
4. Azrue Cloud
5. Sql Server Mangement Studio

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation of Class Diagrams:

Class Diagram of Business Logic shown below:

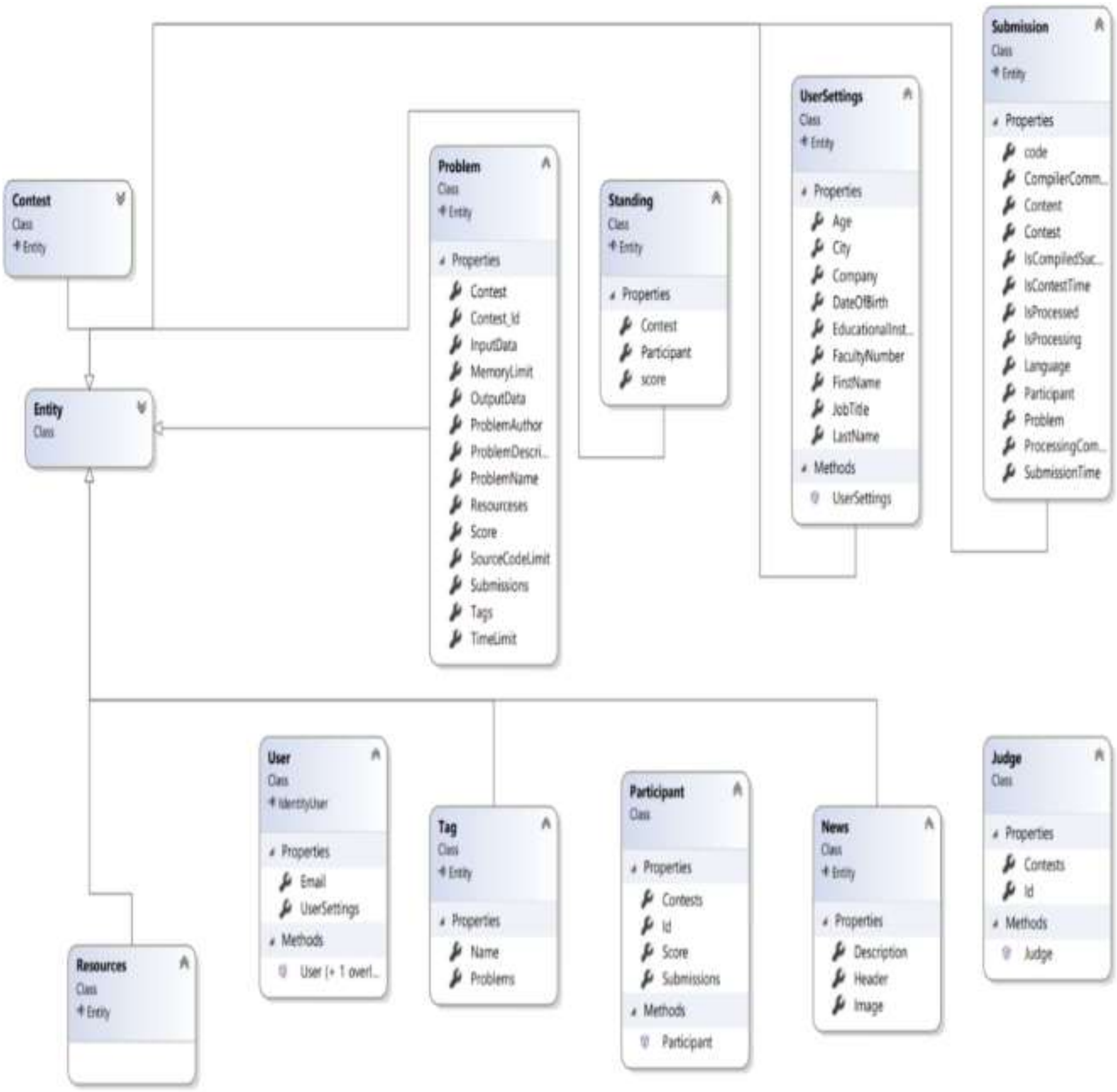


Figure 5.1: Business Logic Class Diagram

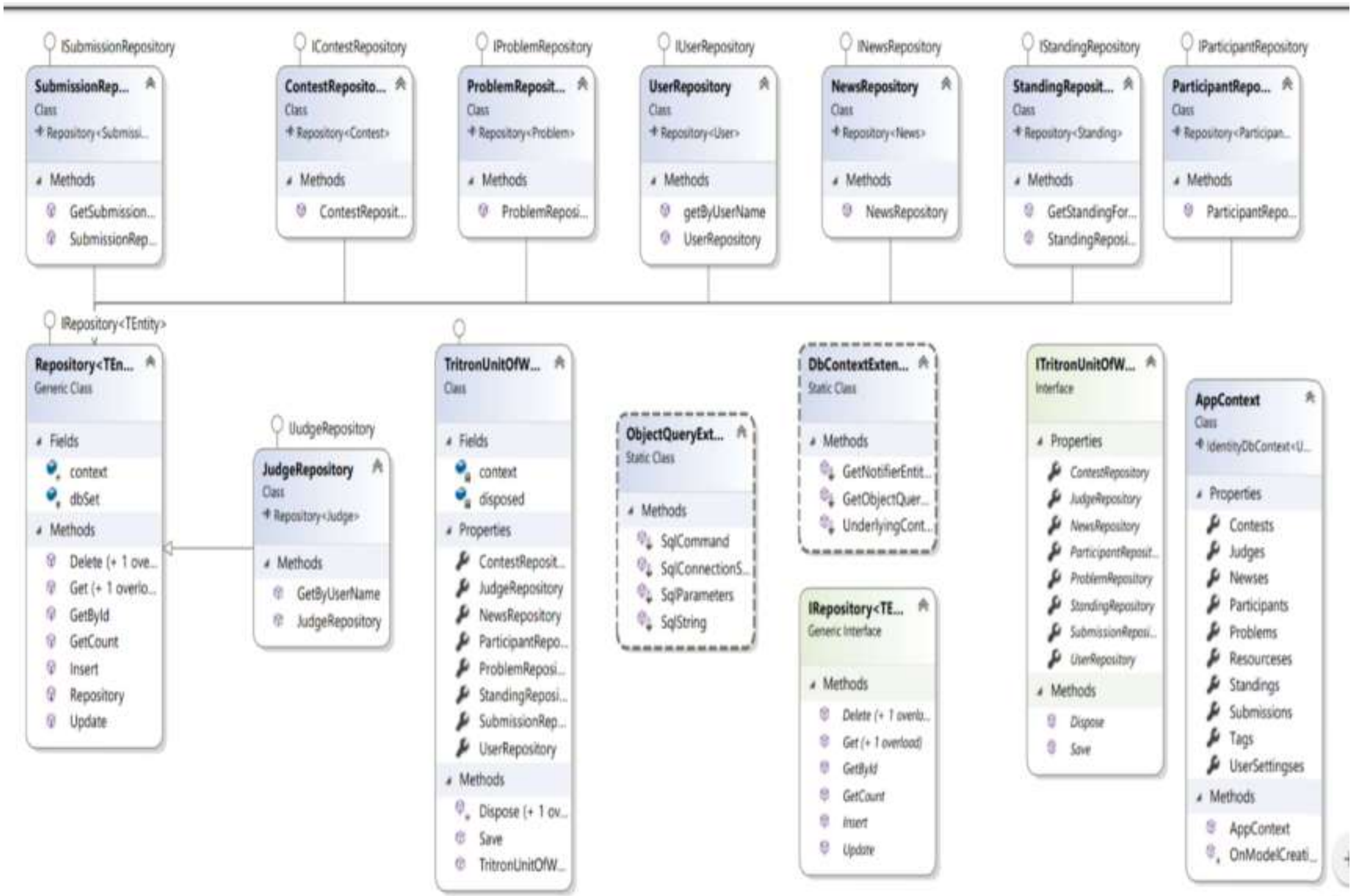

ISubmissionRepository
SubmissionRep...
Class
Repository<Submissi...
Methods
GetSubmission...
SubmissionRep...
IContestRepository
ContestReposito...
Class
Repository<Contest>
Methods
ContestReposit...
IProblemRepository
ProblemReposit...
Class
Repository<Problem>
Methods
ProblemReposi...
IUserRepository
UserRepository
Class
Repository<User>
Methods
getByUserName
UserRepository
INewsRepository
NewsRepository
Class
Repository<News>
Methods
NewsRepository
IStandingRepository
StandingReposit...
Class
Repository<Standing>
Methods
GetStandingFor...
StandingReposi...
IParticipantRepository
ParticipantRepo...
Class
Repository<Participan...
Methods
ParticipantRepo...
IRepository<TEntity>
Repository<TEn...
Generic Class
Fields
context
dbSet
Methods
Delete (+ 1 ove...
Get (+ 1 overlo...
GetById
GetCount
Insert
Repository
Update
IJudgeRepository
JudgeRepository
Class
Repository<Judge>
Methods
GetByUserName
JudgeRepository
TritronUnitOfW...
Class
Fields
context
disposed
Properties
ContestReposit...
JudgeRepository
NewsRepository
ParticipantRepo...
ProblemReposi...
StandingReposi...
SubmissionRep...
UserRepository
Methods
Dispose (+ 1 ov...
Save
TritronUnitOfW...
ObjectQueryExt...
Static Class
Methods
SqlCommand
SqlConnectionS...
SqlParameters
SqlString
DbContextExten...
Static Class
Methods
GetNotifierEntit...
GetObjectQuer...
UnderlyingCont...
IRepository<TE...
Generic Interface
Methods
Delete (+ 1 overlo...
Get (+ 1 overload)
GetById
GetCount
Insert
Update
ITritronUnitOfW...
Interface
Properties
ContestRepository
JudgeRepository
NewsRepository
ParticipantReposit...
ProblemRepository
StandingRepository
SubmissionReposi...
UserRepository
Methods
Dispose
Save
AppContext
Class
IdentityDbContext<U...
Properties
Contests
Judges
Newses
Participants
Problems
Resourceses
Standings
Submissions
Tags
UserSettingses
Methods
AppContext
OnModelCreati...

Figure 5.2: Repository Class Diagram

5.2 Implementation of Front-end Design

Our initial goal was to make a problem solving code judging system. We designed the whole system according to the department's needs. We have a login and sign up page. Without login, one can see the contests list,contest pages and running contest’s standings . After login, users can see the dashboard,contests,problems and some other pages depending on the role of the user.. In the dashboard, we show the count of past tried problem’s verdict and if the user an admin or moderator he can see access to the special pages.In Problem page user see area to submit the problem’s code. Then we implemented standing pages and other systems.

5.3 Implementation of Interactions

The main goal was to shift the algorithmic code judging test to an automated web based system from a pen paper written system. We have made easy to know the result of a code was right or wrong which were impossible before. User can easily assume what to do by seeing the self descriptive webpages

5.4 Testing Implementation

We run our project through some testing procedure to check if it works properly or not. Several tests are presented below to indicate the absolute readiness of the application to function.

Table 5.1: Testing Implementation

No	Tested Case	Tested Input	Expected Outcome	Actual Outcome	Result
1	Sign up	Provide username, email address, password and sign up.	Successfully Registered.	Successfully Registered.	Passed
2	Sign in	Provide an email address, password and	Dashboard comes up if user is	Dashboard comes up if user is	Passed

		sign in.	registered.	registered.	
3	Create contest page	Click on the (+) create contest option.	Go to the create contest page.	Go to the create contest page if admin.	Passed
4	Edit Contest	Create edit button on contest page	Go to edit contest page.	Go to edit contest page. if admin or judge of the specific contest	Passed
5	Submit Problem	Select file from pc and submit button click on problem description page	Submission goes to submission queue and give verdict after few seconds	Submission goes to submission queue if user have access to the problem.	Passed
6	See submission	Go to problem page or see from the contest submission page .and click on the verdict button	Show code if user is admin or the submission was by user himself.	Show submitted code	Passed
7	Show standing	Go to contest page.	Show standing successfully while contest running or ended	Show standing successfully.	Passed

8	Show Contest strating page	Go to contest page	Show all participants in the contest while contest is not started and show countdown timer to show how time rest to start contest	Show countdown	Passed

5.5 Test Results and Reports

5.5.1 Functionality Testing

We run this test to verify our product is as fulfil the specifications and it fulfil the basic requirements as still a few features need to add.

5.5.2 Usability testing

5.5.2.1 Test the site Navigation

Table 5.2: Test the site Navigation

Test Case	**Test Data**	**Result**
Menus	Check in different device	Easily visible and consistent
Buttons	Check in different device	Easily visible and consistent
Links	Check in different device	Consistent

5.5.2.2 Test the Content

Table 5.3: Test the Content

Test Case	**Test Data**	**Output**

Content	Readable content without errors	Passed
Images	contain an "alt" text	Passed

5.5.3 Interface Testing

In this section we tested Database Server, Web Server and Application.

5.5.3.1 Application

Table 5.4: Application

Test Case	**Test Data**	**Output**
From Client Side	Test request correctly sent to the server	Passed
To Client Side	The data on the client-side is displayed accurately	Passed
Error in Application	Should be shown to the administrator only	Passed
Error in Application	Should not be shown to the administrator only	Failed

5.5.3.2 Database Server

Queries sent to the database resulting in expected results, but creating error in some cases.

Table 5.5: Database Server

Test Case	**Test Data**	**Output**
Queries	Sent to the server with the expected output	Passed
Queries	Error free	Passed

5.5.4 Database Testing

Table 5.6: Database Testing

Test Case	**Test Data**	**Output**
Data Integrity	Check create database, Check update database, Check delete database.	Passed
Response time	Checking response time if it is good	Passed
Data	Data is retrieved properly.	Passed
Error	Error free	Passed

5.5.5 Crowd Testing

Table 5.7: Crowd Testing

Test Case	**Test Data**	**Output**
Application	Response times at different connection speeds	Passed
Crash occurrence	Does Not occur Due to peak load	Passed(three programming contest with around 200 students participation in each)

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

Tritron Online Judge empowers not only in programming contest sector but also in taking lab evaluation and can be used in academic sector and as well as software engineer recruitment. It's barrier less access to set algorithmic problem in the website will help to grow problem setters and solvers community among the students.

6.2 Scope for Further Developments

As Tritron Online Judge app in its initial phase it does have lots of area to develop. We have plan to move the frontend in a single page application that will give users mobility and better UX experience. In the meantime, an online code editor can be integrated with this software which will give users the advantage of the flexibility to not use any IDE in their personal computer and use our online resource for coding.

APPENDIX

APPENDIX A: PROJECT REFLECTION

Implementing this project, we introduced with a number of new and updated technology and system which left us with excellent experience and grown our thirst to learn more technology and work with more web-based work. In brief this whole journey was fantastic and helping and working altogether made us complete our project this much.

References:

[1] CodeMarshal, available at <<https://algo.codemarshal.org/>>, last accessed on 16-10-2019 at 10:00 PM.

[2] HackerRank, available at <<https://www.hackerrank.com/home>>, last accessed on 16-10-2019 at 10:50 PM.

[3]LightOj , available at<<http://lightoj.com/login_main.php>>,last accessed on 16-10-2019 at 10:50 PM.

[4] HTML, available at <<https://en.wikipedia.org/wiki/HTML>>, last accessed on 19-10-2019 at 08:00 PM.

[5] Cascading Style Sheets, available at <<https://en.wikipedia.org/wiki/Cascading_Style_Sheets>>, last accessed on 19-10-2019 at 08:20 PM.

[6] Why do we use Bootstrap?, available at <<https://www.quora.com/Why-do-we-use-Bootstrap>>, last accessed on 19-10-2019 at 08:40 PM.

[7] JavaScript, available at <<https://en.wikipedia.org/wiki/JavaScript>>, last accessed on 19-10-2019 at 09:10 PM.

[8] What's New in ASP.NET MVC 5 that make your MVC web sites shine, available at <<https://www.dotnetcurry.com/aspnet-mvc/975/new-features-aspnet-mvc-5>>, last accessed on 19-10-2019 at 10:15 PM.

[9]EntityFrameowork?,availableat<<https://docs.microsoft.com/en-us/ef/>>, last accessed on 19-10-2019 at 11:00 PM.

[10] SignalR, available at <<https://dotnet.microsoft.com/apps/aspnet/signalr>>, last accessed on 19-10-2019 at 11:30 PM.