Project Report

Online Competitive Programming Portal

[*Ground-Zero*]

2016

[Aditya Pratap Singh]
[136177847]
[Project Guide]
[Mr. Ashish Shukla]

ACKNOWLEDGEMENT

"It is not possible to prepare a project report without the assistance & encouragement of other people. This one is certainly no exception."

On the very outset of this report, I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavor.

Without their active guidance, help, cooperation & encouragement, I would not have made headway in this project.

I am ineffably indebted to my project-guide ***Mr. Ashish Shukla*** *for his conscientious guidance and encouragement to accomplish this project.*

I also acknowledge with a deep sense of reverence, my gratitude towards my parents and members of my family, who has always supported me morally as well as economically.

At last but not least, gratitude goes to all of my friends who directly or indirectly helped me to complete this project report.

Any omission in this brief acknowledgement does not mean lack of gratitude.

[Aditya Pratap Singh]

ONLINE COMPETITIVE PROGRAMMING PORTAL

TABLE OF CONTENTS

SYNOPSIS

Introduction:

Purpose and Objective:-

The *Online Competitive Programming Portal* is developed with the objective to allow students of Computer Science to practice and hone their programming skills. It provides an opportunity to students and individuals to practice programming at any time of the day. This web-app also allows it's user to compare himself/herself to other users as the results and ranks of other users are published under a common area.

Project Category:-

The project is a web-based application working on a Relational Database Management System – architecture.

Scope:-

This project is comparatively much broad and productive than the usual way of practicing programming. The scope of this project could be summarised in the following points:

- It can be used by Universities and Private Educational Institutes to conduct programming contests and mock tests.

- Being a web-based application the user can access the portal from any part of the world.
- Does not require an examiner to be physically present with the user thereby greatly increasing portability.
- Helps users to achieve greater productivity in lesser time.

Document Conventions:-

This document is prepared using the IEEE's recommended practice for Software Requirement Specification.

Definitions, Acronyms and Abbreviations:-

The various acronyms and abbreviations found throughout the project:

- H.T.M.L. => Hyper Text Mark-up Language.
- C.S.S. => Cascading Style Sheet.
- J.S.P. => Java Server Pages.
- O.C.P.P => Online Competitive Programming Portal.
- J.2E.E. => Java To Enterprise Edition.
- S.Q.L. => Structured Query Language.

Technologies To Be Used:-

Various technologies which will be used during the implementation phase of this project are:

- **H.T.M.L.:** Application Design.
- **C.S.S.:** Application Style and Formatting.
- **JavaScript:** Scripts for validations and small tweaks.
- **J.2E.E. (J.S.P., Servlet , JavaBeans):** Application Architecture.
- **MySQL:** Database Architecture.

Overview:-

This S.R.S. document will include primarily two sections, they are:

Overall Description: Major components of the system, interconnections, and external interfaces will be described under this section.

Specific Requirements: The constraints and limitations of the system, the role of the user of the web-app will be defined in this section.

References:-

The project is prepared with references from various sources like *STACKOVERFLOW*, *STACKEXCHANGE* and *IEEE's* website.

Overall Description:

Product Perspective:-

O.C.P.P. as in Online Competitive Programming Portal is developed with the perspective to be *open-source*, under the *GNU General Public License*. This web-based application depicts a perfectly capable Client-Server model allowing it's users to practice programming problems. This project helps it's users to sharpen-up their coding skills in a virtual environment in absence of a guide or a teacher.

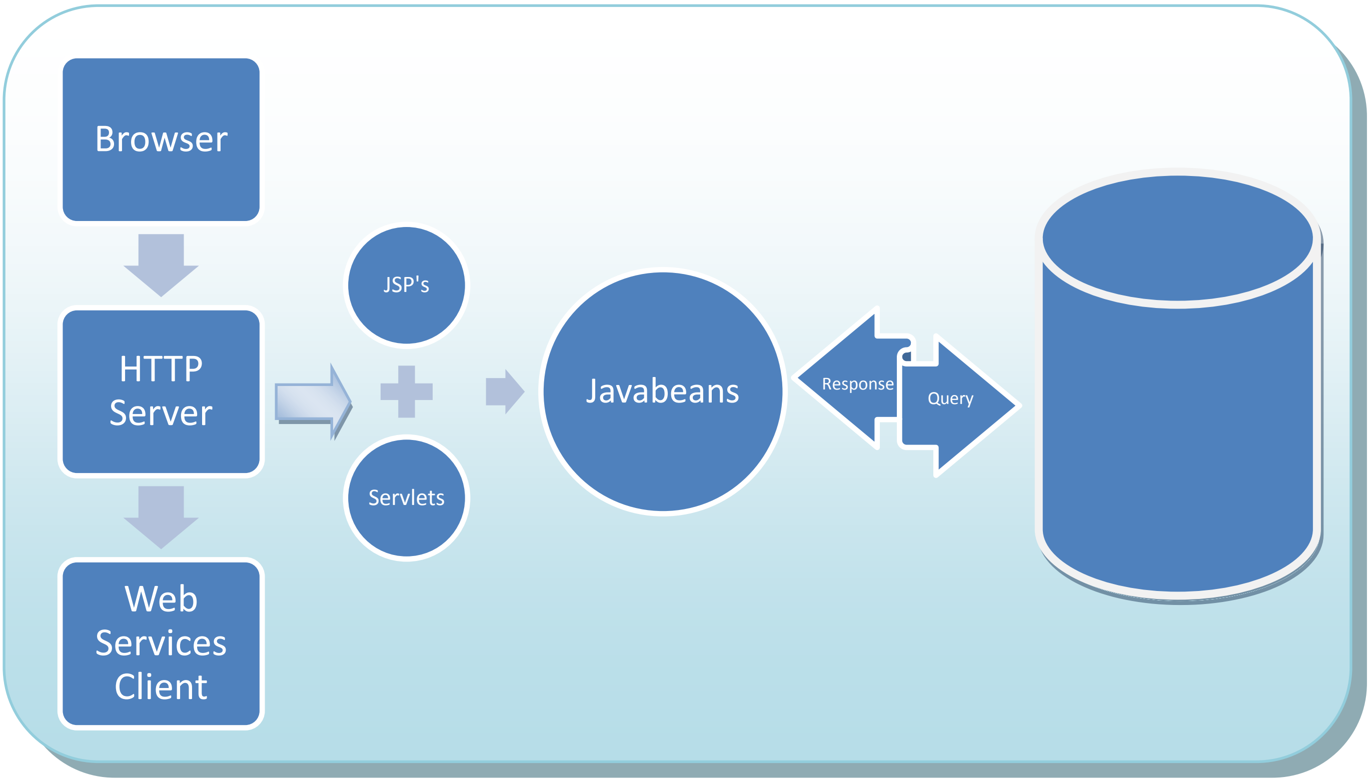


FIGURE: PRODUCT PERSPECTIVE

Software Interface:-

The software requirements for the project to work efficiently are listed as follows:

- **FRONT-END (CLIENT-SIDE):**
  - Any Windows or Linux based operating system.
  - A working internet connection.
  - An internet browser, preferably supporting H.T.M.L.5 and C.S.S.3.
- **WEB-SERVER:**
  - A Linux or UNIX environment with Apache Tomcat.
  - Although any latest web-server could host the project but Apache Tomcat is recommended for optimum performance.
- **DATABASE SERVER:**
  - The project uses MYSQL as its Database.

Hardware Interfaces:-

The minimum hardware requirements for the project to be hosted and used are:

- **CLIENT SIDE:**
  - **MINIMUM REQUIREMENTS:**
    - Intel Atom or Celeron Processor with

        - 128 Mega-Bytes of R.A.M.
        - A working internet connection.

- **SERVER SIDE:**
    - **MINIMUM REQUIREMENTS:**
        - Intel Pentium Processor with
        - 512 Mega-Bytes of R.A.M.
        - 1.5 G.B. Hard Disk for project source and web-server.
        - Storage space for Database according to database size.
    - **RECOMMENDED REQUIREMENTS:**
        - Intel Multi-Core Processor with
        - 1 Giga-Byte of R.A.M.
        - 1.5 G.B Hard Disk for project source and web-server.
        - Storage space for Database according to database size.

Product Schedule:-

The product is scheduled to be completed within 2.8 months or around 90 days. A complete chart of the process or activities to be done is given below which is divided into two parts,

- *Project Evaluation & Review Technique* chart.
- *GANTT* chart (Named after Henry L. Gantt).

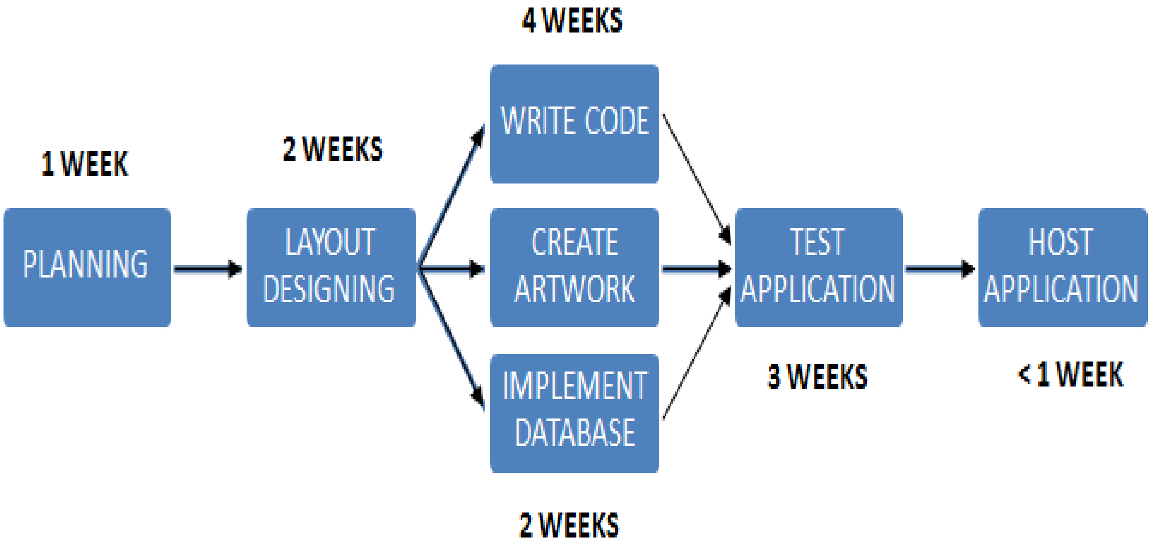


FIGURE: PERT CHART

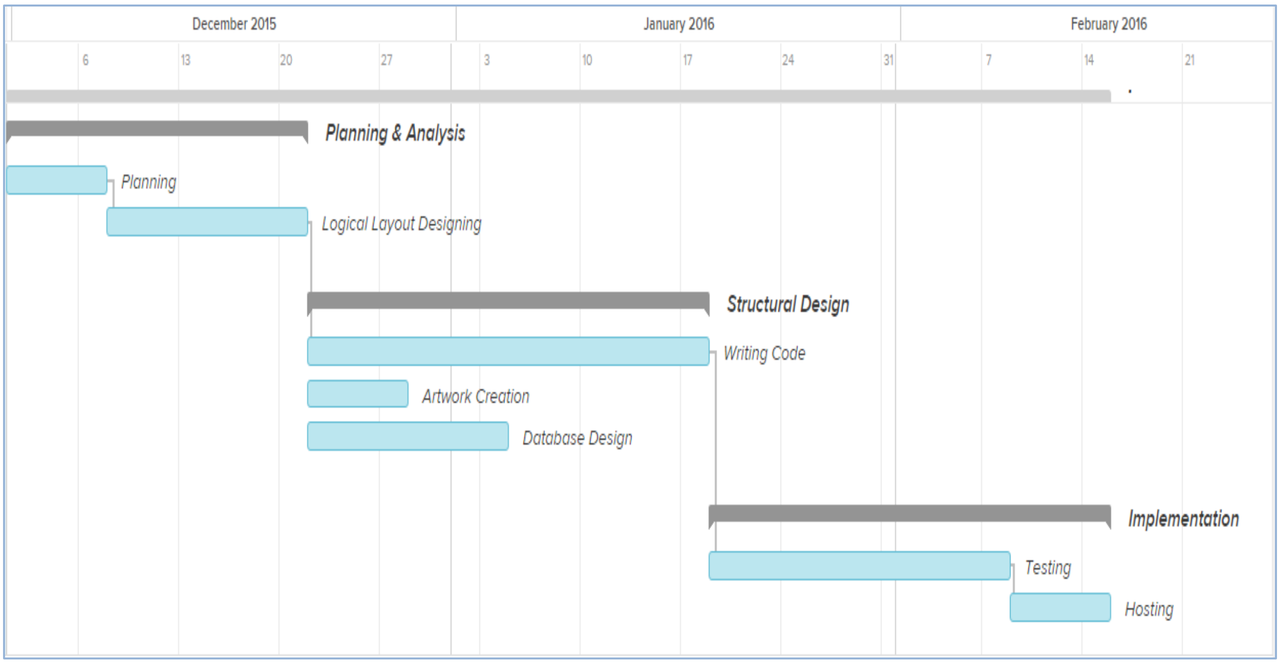
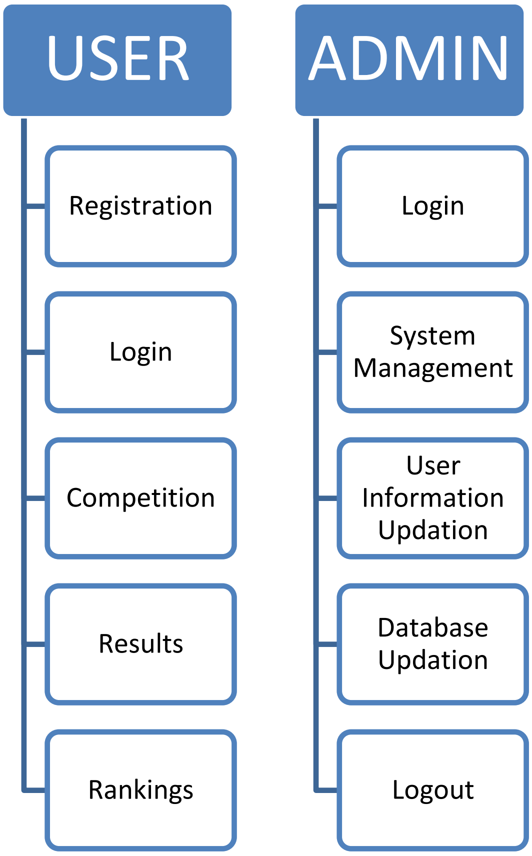


FIGURE: GANTT CHART

Product Functions:-

This application allows its registered users to enter the ‘Competition Module’ and start practicing on programming problems. Various *Modules* of the project are described as follows:



REGISTRATION MODULE: This module will allow a first-time visiting user to register his/her credentials into the portal’s database. Upon successful registration they will be redirected to the login module where they can provide their credentials to enter the competition module.

LOGIN / LOG-OUT MODULE: This module will also be divided under two parts, i.e. User and Admin. On the basis of some pre-defined logic the user will be directed to either user's area or admin area based on their login credentials.

COMPETITION MODULE: This module will give access to the competition area to its users. The difficulty level of the competition will be divided into three modes – Easy, Intermediate or Expert. The user will be given a program which will be fetched from the database on the fly. The user will then have to type the program in their preferred language in the provided area and submit the same.

RESULTS MODULE: This module will provide the result of the user currently logged-in. The result will be presented under three categories of the programs the user has attempted i.e. Easy, Intermediate & Expert.

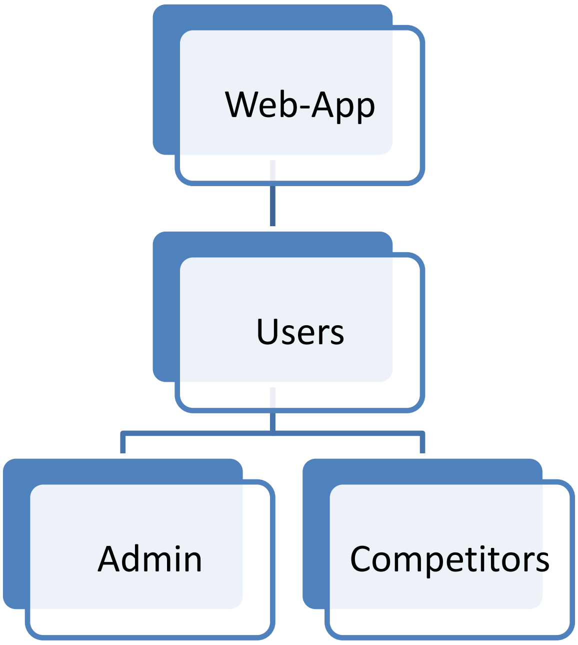
RANKING MODULE: Under this module the Name and Rank of top ten scorers will be displayed. The module will only be accessible by logged-in users.

MANAGEMENT MODULE (ADMIN): This module is specifically for administrators only. A normally registered user will not be able to access this area. Upon logging-in the admin could then manage user-profiles, results and rankings of the users.

MANAGEMENT MODULE (USER): Under this module a registered user could manage his/her profile, delete his account etc.

User Characteristics:-

The users interacting with the system is divided into two sub-classes:



ADMINISTRATORS: The admin will interact with the system to manage the database and user profiles.

USERS (COMPETITORS): These are the users who will be taking part into the programming competition and checking their results & rankings.

Constraints:-

This application is constrained with a properly running internet connection. Since it fetches data from the database server over the internet, it is crucial that a proper internet connection is maintained at all times for the application to function.

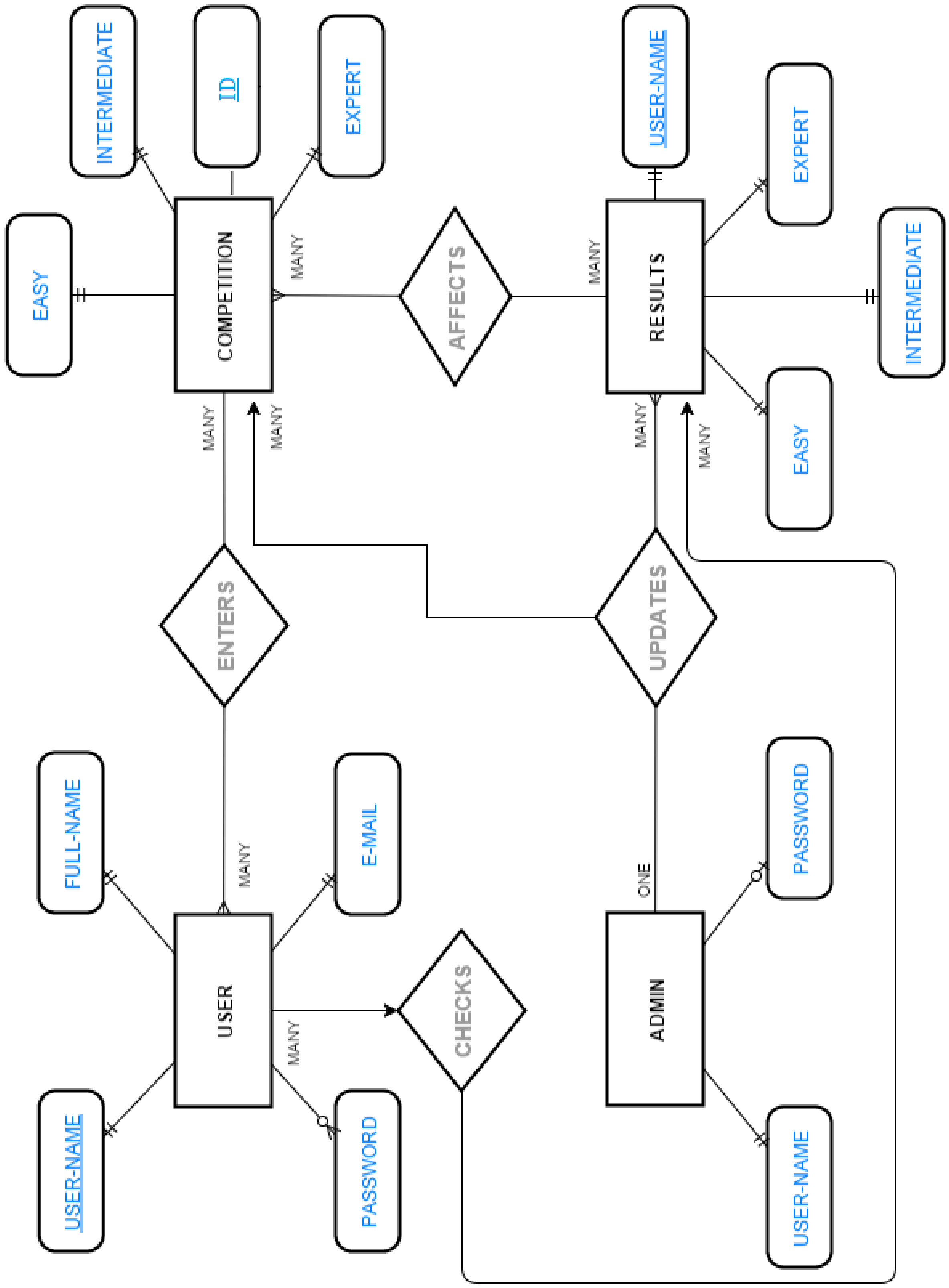
Also the web-portal will also be constrained by the capacity of the database. Since it's a multi-user platform the database may queue-up incoming requests if it exceeds a certain limit.

Assumptions and Dependencies:-

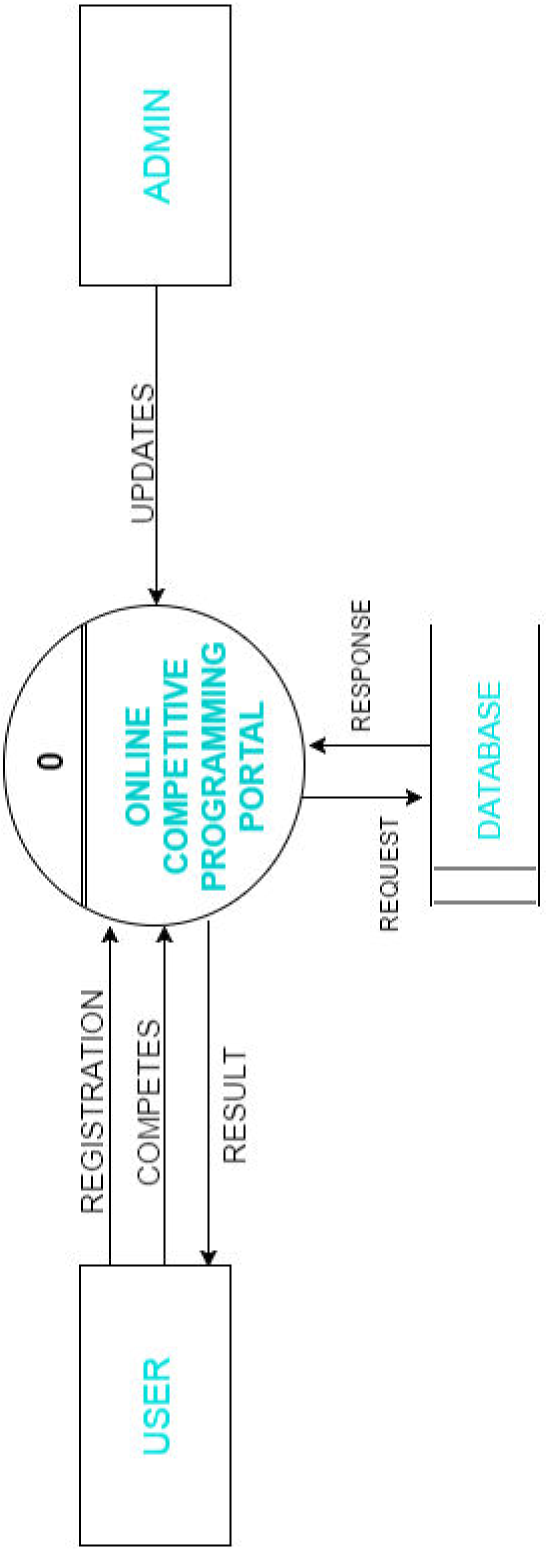
Proper working of this web-app is dependent on the internet connectivity of the user's computer.

- It is assumed that the user has a basic knowledge of operating a computer system.
- It is assumed that the user does not cheat during the competition time.

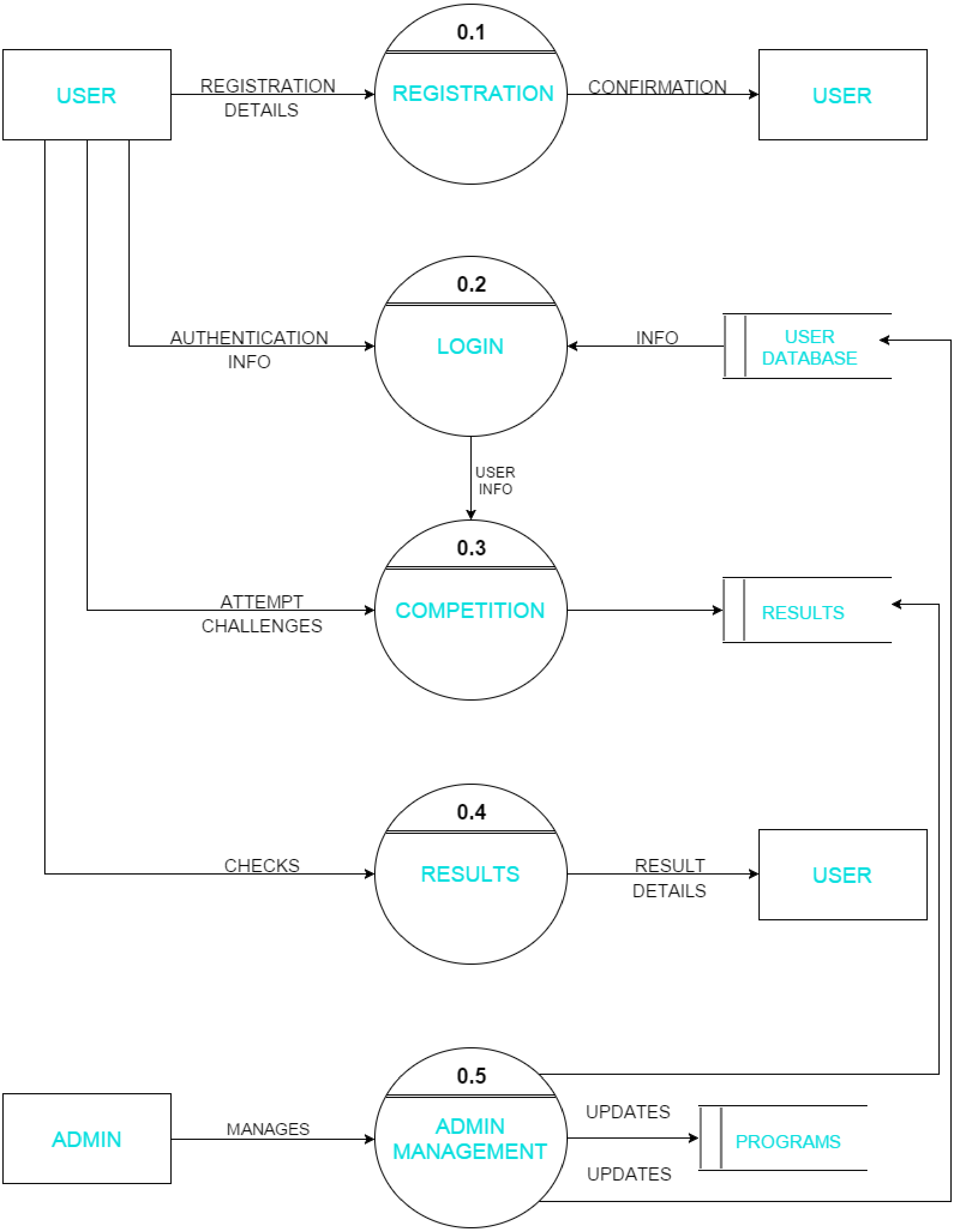
Entity Relationship Diagram:-



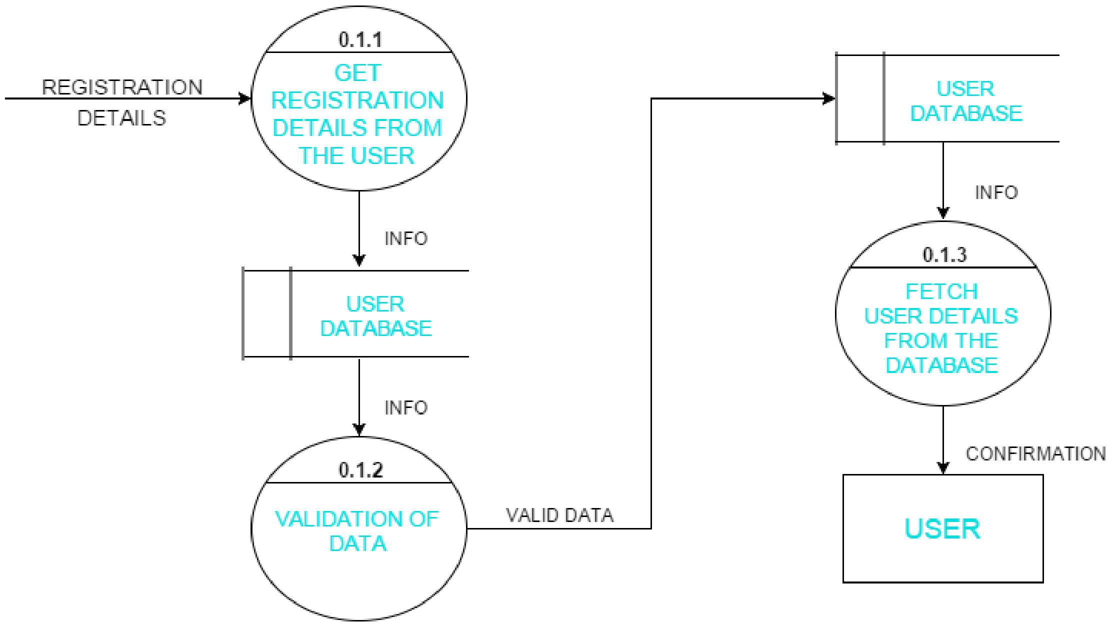
Data Flow Diagram:-



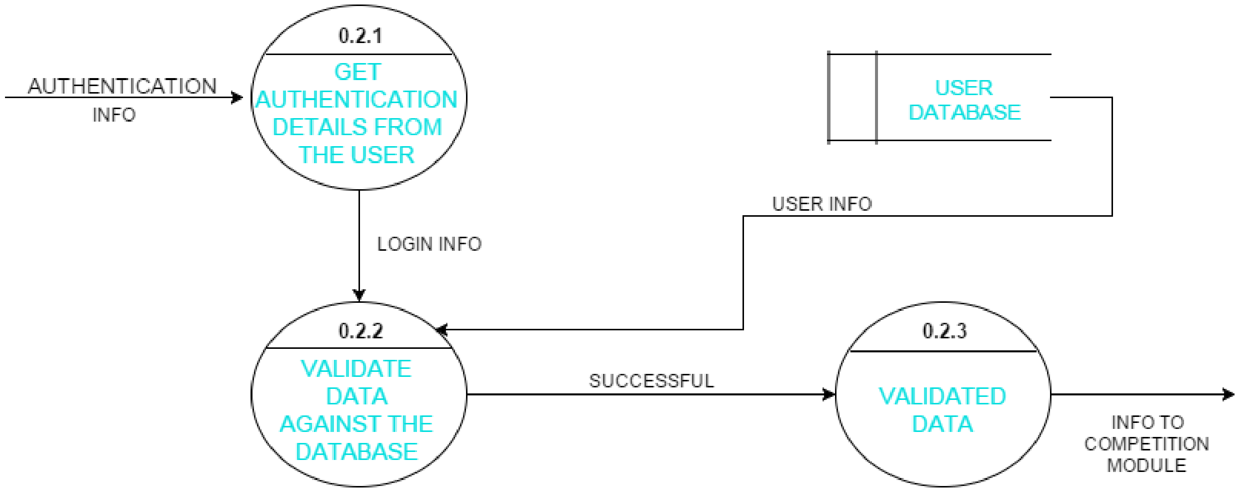
LEVEL- 0
CONTEXT LEVEL
DATA FLOW DIAGRAM



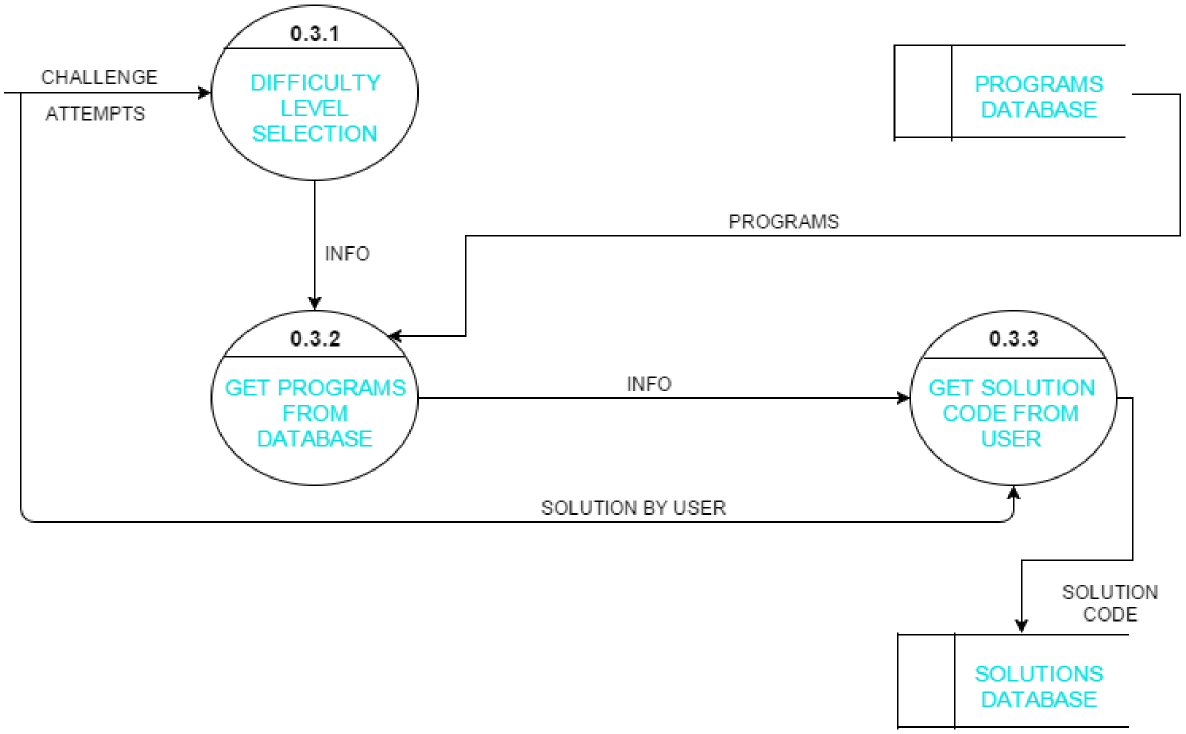
LEVEL- 1
DATA FLOW DIAGRAM



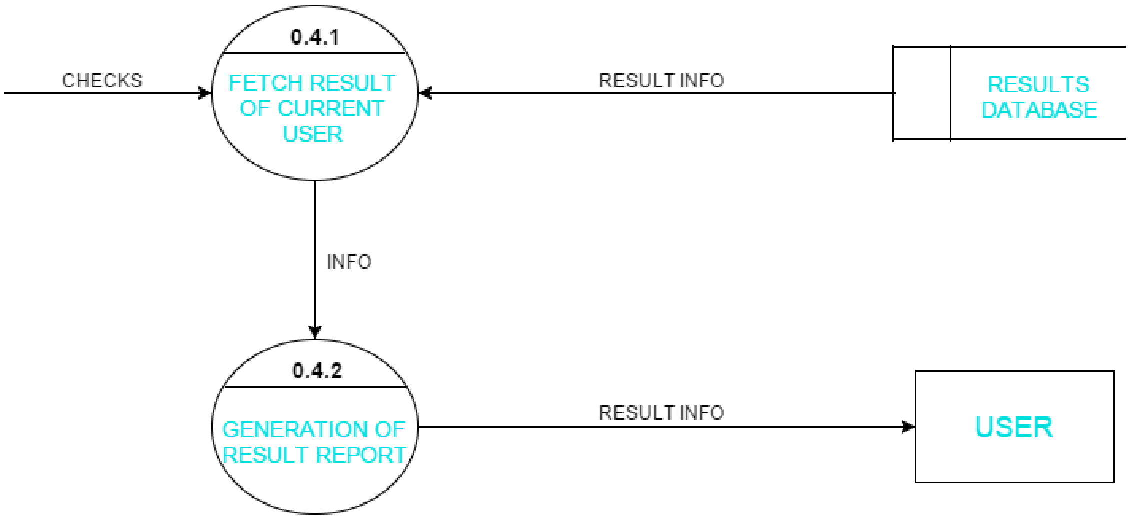
LEVEL - 2
REGISTRATION MODULE
DATA FLOW DIAGRAM


AUTHENTICATION INFO
0.2.1
GET AUTHENTICATION DETAILS FROM THE USER
LOGIN INFO
USER DATABASE
USER INFO
0.2.2
VALIDATE DATA AGAINST THE DATABASE
SUCCESSFUL
0.2.3
VALIDATED DATA
INFO TO COMPETITION MODULE

LEVEL - 2
LOGIN MODULE
DATA FLOW DIAGRAM



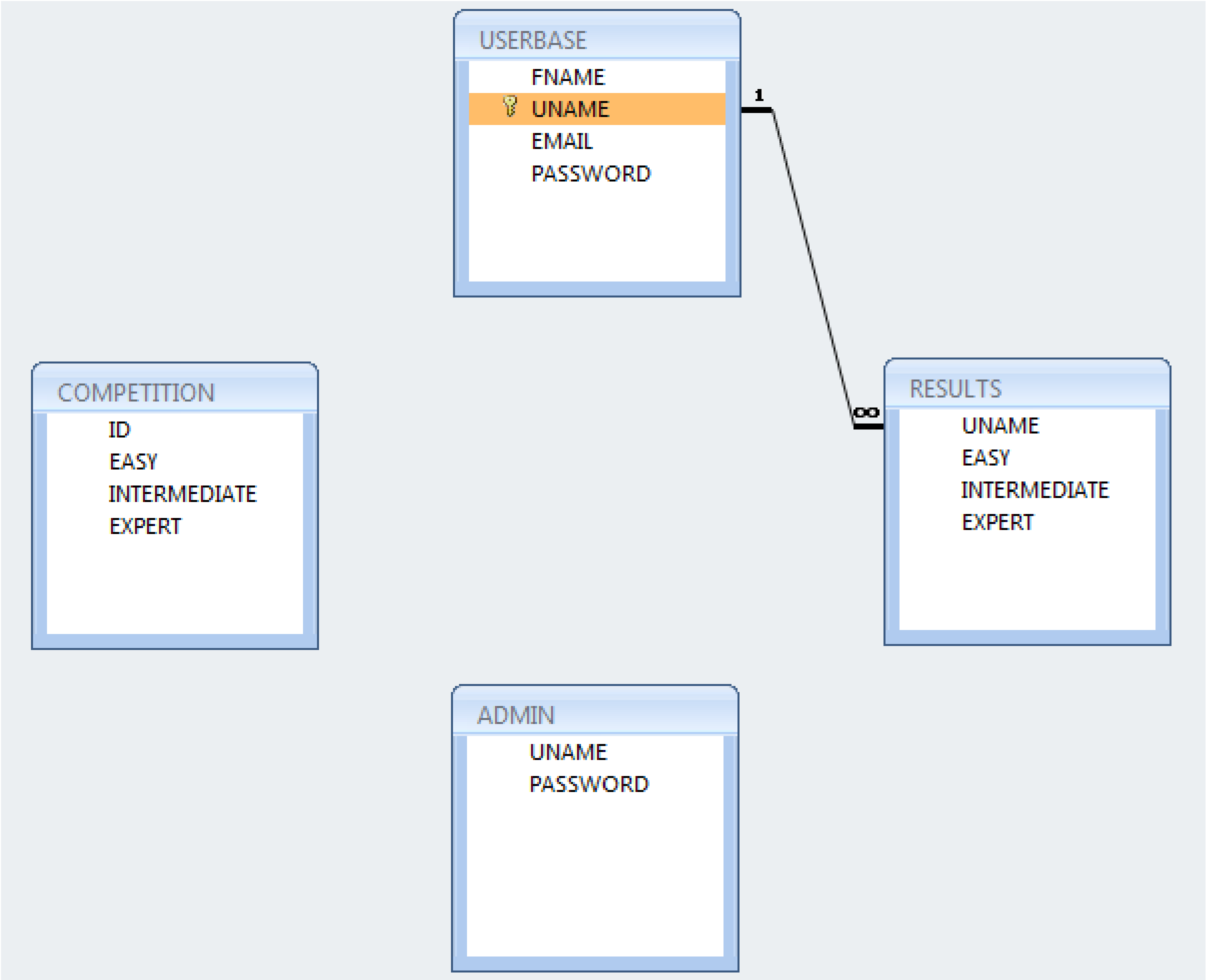
LEVEL - 2
COMPETITION MODULE
DATA FLOW DIAGRAM



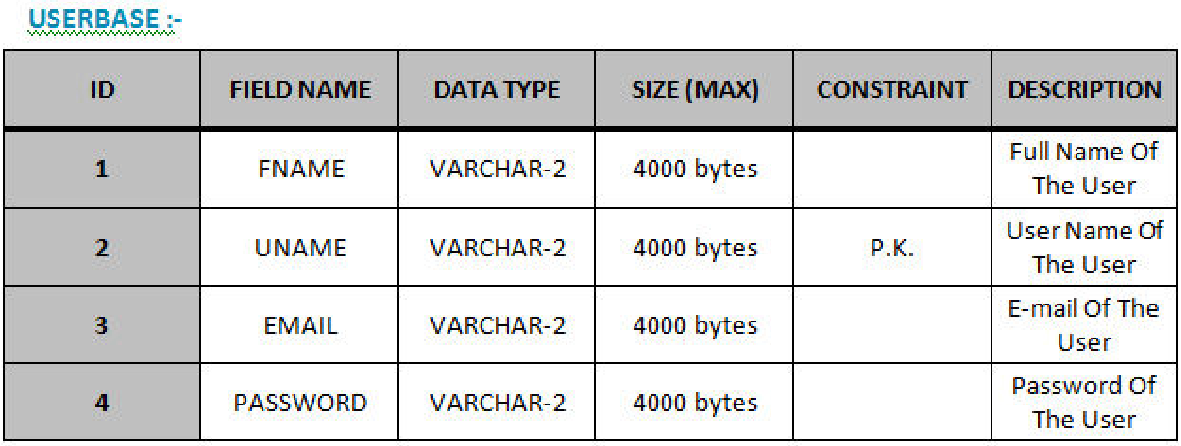
LEVEL - 2
RESULTS MODULE
DATA FLOW DIAGRAM

Database Design:-

Following figure illustrates the database design of the project (Tables and Relationship between them).

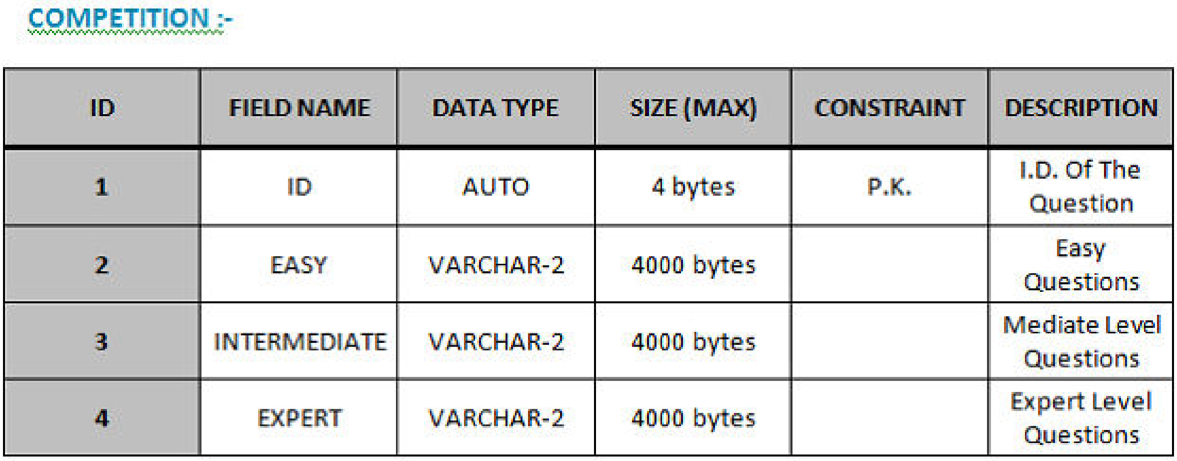


Data Dictionary :-



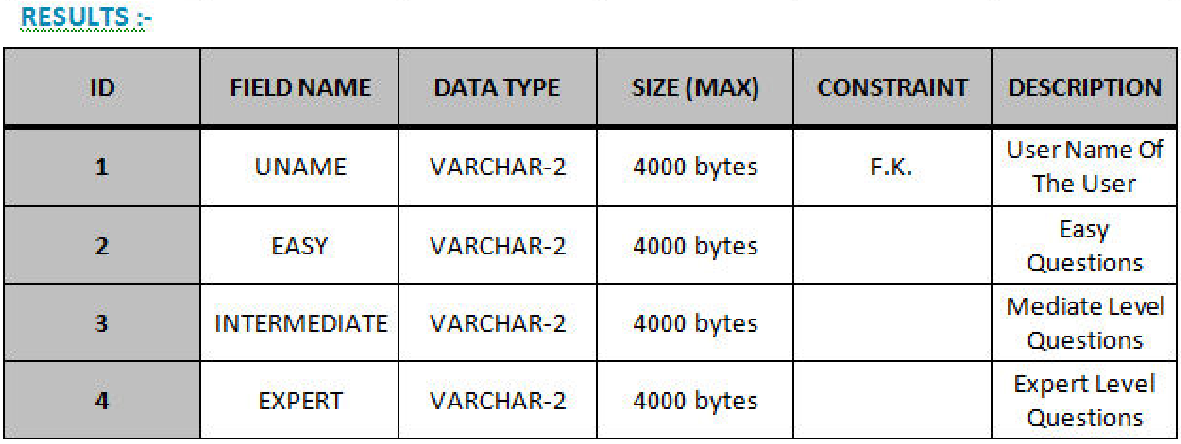
USERBASE :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	FNAME	VARCHAR-2	4000 bytes		Full Name Of The User
2	UNAME	VARCHAR-2	4000 bytes	P.K.	User Name Of The User
3	EMAIL	VARCHAR-2	4000 bytes		E-mail Of The User
4	PASSWORD	VARCHAR-2	4000 bytes		Password Of The User



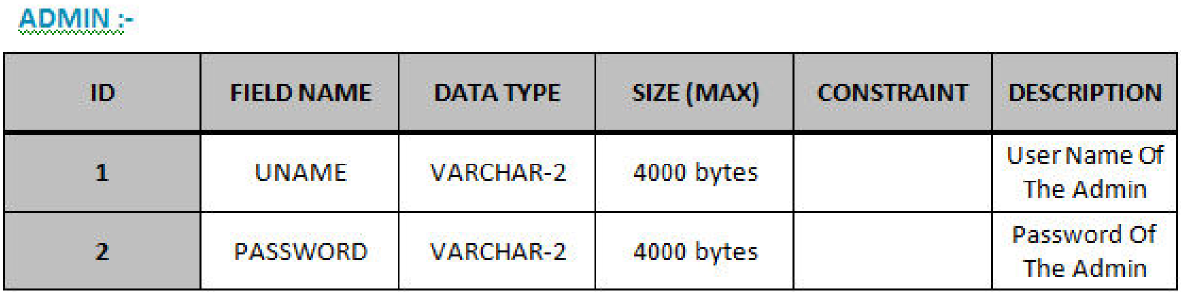
COMPETITION :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	ID	AUTO	4 bytes	P.K.	I.D. Of The Question
2	EASY	VARCHAR-2	4000 bytes		Easy Questions
3	INTERMEDIATE	VARCHAR-2	4000 bytes		Mediate Level Questions
4	EXPERT	VARCHAR-2	4000 bytes		Expert Level Questions



RESULTS :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	UNAME	VARCHAR-2	4000 bytes	F.K.	User Name Of The User
2	EASY	VARCHAR-2	4000 bytes		Easy Questions
3	INTERMEDIATE	VARCHAR-2	4000 bytes		Mediate Level Questions
4	EXPERT	VARCHAR-2	4000 bytes		Expert Level Questions



ADMIN :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	UNAME	VARCHAR-2	4000 bytes		User Name Of The Admin
2	PASSWORD	VARCHAR-2	4000 bytes		Password Of The Admin

Specific Requirements:

Performance Requirements:-

- Loading speed of all web-pages should be under 2 to 3 seconds.
- Any operation based on Database should not take more than 5 seconds.

Design Constraints:-

The O.C.P.P. project must adhere to the following standards or protocols:

- Web pages are to be designed using HTML-5 and formatted using CSS-3 transitional standards.
- W3C Web Accessibility standards, Web Content Accessibility Guidelines, should be followed including keyboard navigation, alternate titles for images, etc.

Software System Attributes Requirements:-

Reliability:-

The database operations should be completed within 5 seconds and must be reliable more than 99.5% of time.

Availability:-

The portal should be designed to be available no less than 99.99% of time. All upgrades to system including software updates, patches and fixes must be done without taking the portal offline.

Security:-

Security standards to be followed are:

- User Id's and Passwords of users must be hashed using one-way hashing algorithms before storing them into databases.
- Appropriate methods like *CAPTCHA's* should be implemented to prevent machine brute-force attacks.
- Any input by user must be validated and sanitized for SQL injection scenarios.
- A well-defined password policy is to be adopted including password-change frequency, invalid attempts, etc.

Maintainability:-

- Every code module should have a well documented manual including logic.

- All code components should be thoroughly tested and the test coverage should be more than 80%.

Future Scope:-

Currently the project is a standalone application developed from scratch but in future with a wider range of audience the project can be ported to a professional industry-standard framework. Performance of the project can also be improved by implementing the project on a multi-server platform.

Also, currently the software only supports manual checking of the user-submitted solution codes. This in future could be automated for a number of programming languages, as required.

Project Ownership:-

This project is ***NOT*** done for any client/ industry, its complete ownership lies with the *Project Creator*.

PROJECT REPORT

INTRODUCTION

Purpose:-

Online Competitive Programming Portal, a web-based application has been developed for students who want to practice programming. It provides an easy to use interface which they can utilize to hone their skills without the need of a teacher to cross-check their work.

The portal essentially allows a student to tackle programming problems right from their homes, or for that matter from anywhere around the world at their own leisure.

Document Conventions:-

The document follows IEEE's recommended guidelines for Software Requirement Specifications.

Intended Audience:-

The intended audience could be any individual who wish to practice programming or a small institute or college which could use this web-app to conduct mock-tests or programming contests.

Project Scope:-

The project's scope is comparatively much wider than the conventional way of programming practice.

- It completely eliminates the use of paper.
- An institute which conducts manual contests can use this web-app to automate the whole process.
- An examiner or invigilator is not required if and when the student attempts the questions.
- Saves travelling time/cost for students.
- Elimination of manual work of preparing and storing results.
- Being web-based it can be used from any part of the world at any time.

References:-

The project is prepared by researching various resources like IEEE's website, IGNOU's website etc.

PROJECT DESCRIPTION

Perspective:-

O.C.P.P. as in Online Competitive Programming Portal is developed with the perspective to be *open-source*, under the *GNU General Public License*. This web-based application depicts a perfectly capable Client-Server model allowing it's users to practice programming problems. This project helps it's users to sharpen-up their coding skills in a virtual environment in absence of a guide or a teacher.

Functions:-

The application allows its users with the following functionalities:

- Registering their account on the portal (Registration Module).
- Authentication before entering a competition (Login Module).
- Competing with students/individuals from around the world (Competition Module).
- Viewing results of previously attempted contests/competitions (Results Module).
- View rankings of all users who are registered on the portal.
- Allowing an admin to update the app's database (Admin Module).

Modules:-

Project's different modules and their description are as follows:

REGISTRATION MODULE: This module allows an individual to register themselves to the portal and start competing right away. The module is designed in such a way that it doesn't require the user to fill up lengthy unnecessary forms. Only necessary information is asked during this process.

AUTHENTICATION MODULE (LOGIN): The authentication or login module acts as a barrier between unauthorized or non-registered users and the portal to refrain them from gaining access to the competition area or other users information.

COMPETITION MODULE: This module presents users with three choices viz. Easy, Intermediate and Expert. These three choices are the level of hardness of programming problems the users will encounter. Under this same module the user is expected to write their solution code according to the problem and submit the same.

RESULTS MODULE: The results module will display the points acquired by a user in previous competitions/contests. The results will also be divided into three categories – Easy, Intermediate and Expert.

RANKINGS MODULE: This module will display the rankings of every user who is registered onto the portal. The rankings are determined by total points accumulated by a user.

ADMIN MODULE: The admin module is designed to update the database of the application. The admin module is divided into three sub-modules – they are:

1. **Manage Users –** Under this sub-module the admin can delete a user if so required.
2. **Update Problems –** This sub-module can be used by the admin to introduce new problem statements into the applications database.
3. **Update Results –** An admin can access this area to update results of any user he wishes.

Though the admin must authenticate himself/herself with the portal to gain rights to these sub-modules. A user who is registered through the portal itself cannot access these modules.

Operating Environment:-

As this is a web-based application, it can perform well within its criteria on any operating system with an internet connection and a web browser.

User Classes:-

The users of this application can be categorized into two major groups, they are –

ADMINISTRATORS - An admin will be responsible for the management of the app and its database. He can log-in with his credentials to make the required changes as and when necessary like updating results, adding new problems etc.

INDIVIDUALS – An individual such as a student or a programming enthusiast is the main targeted audience for this app. These types of users, unlike the admin can register themselves through the portal itself and participate in the competitions.

Design Constraints:-

The application is designed in such a way that a user can attempt one programming problem only once in a session. To solve the same question another time the user must refresh their session (Must log-out and log-in again).

User Documentation:-

The project will include proper documentation. It will include product overview, complete configuration information of the required hardware and software, technical details and contact information for maintenance purposes.

Assumptions & Dependencies:-

Proper working of this application is totally dependent on user’s internet connection. Different assumptions which are made are:

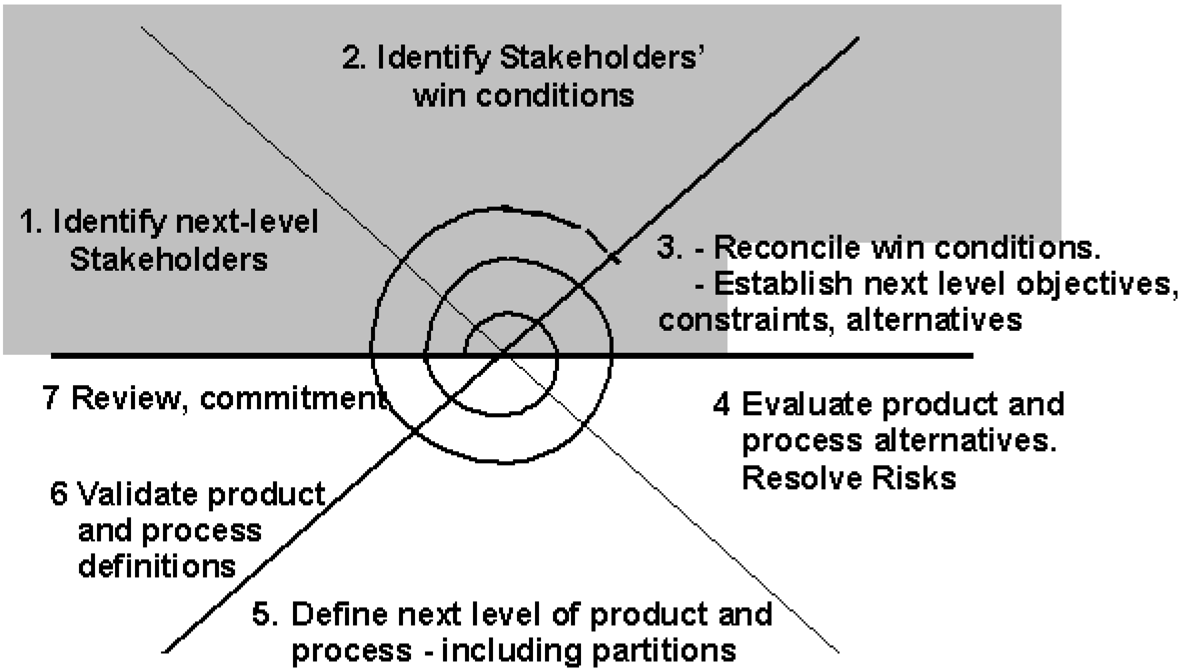
- The user has basic knowledge of working on a web-based application.
- The data entered during registration by the user is completely true.
- The user does not cheat during competitions.

ANALYSIS

Software Design Model:-

The software design model used in developing this web based application is 'Win-Win Spiral' model.

The main reason for using this model is that the developer can push out a basic version of the software as soon as possible and then keep working on adding additional features.


2. Identify Stakeholders' win conditions
1. Identify next-level Stakeholders
3. - Reconcile win conditions.
- Establish next level objectives, constraints, alternatives
7 Review, commitment
4 Evaluate product and process alternatives. Resolve Risks
6 Validate product and process definitions
5. Define next level of product and process - including partitions

WINWIN Spiral Model

Under this model the first version of the software contains features that are of only very high priority. As this project is strictly time-bound 'Win-Win Spiral' appears to be the best choice.

Feasibility Study:-

To assess the feasibility of this application portal, a comprehensive study was made with following conclusions –

OPERATIONAL FEASIBILITY – The application will be very feasible under this category as it will completely eliminate the use of manual programming tests, contests etc. It will also remove the requirement of manual preparation and management of results.

TECHNICAL FEASIBILITY – Most institutes wanting to deploy this application should already be equipped with a computer lab where one of the computers could act as a server for the application as it requires minimum specification hardware.

SOCIAL FEASIBILITY – Most of the users who will be registering to the portal would already be having the basic understanding of working on a web based application.

ECONOMIC FEASIBILITY – As most of the hardware required will already be available, the project will be very cost effective to deploy. On the other hand it will also save the cost of overhead expenses like paper sheets and other stationary items.

MARKET RESEARCH – Market research indicates that this project will be very beneficial to a large section of students who are the targeted audience for it.

ALTERNATIVE SOLUTION – An alternative to the bespoke system could be a database system. This will provide all-around the same functionality internally (institute/college) but will not allow competitions to take place online.

HARDWARE/SOFTWARE REQUIREMENTS

Software Interface:-

The software requirements for the project to work efficiently are listed as follows:

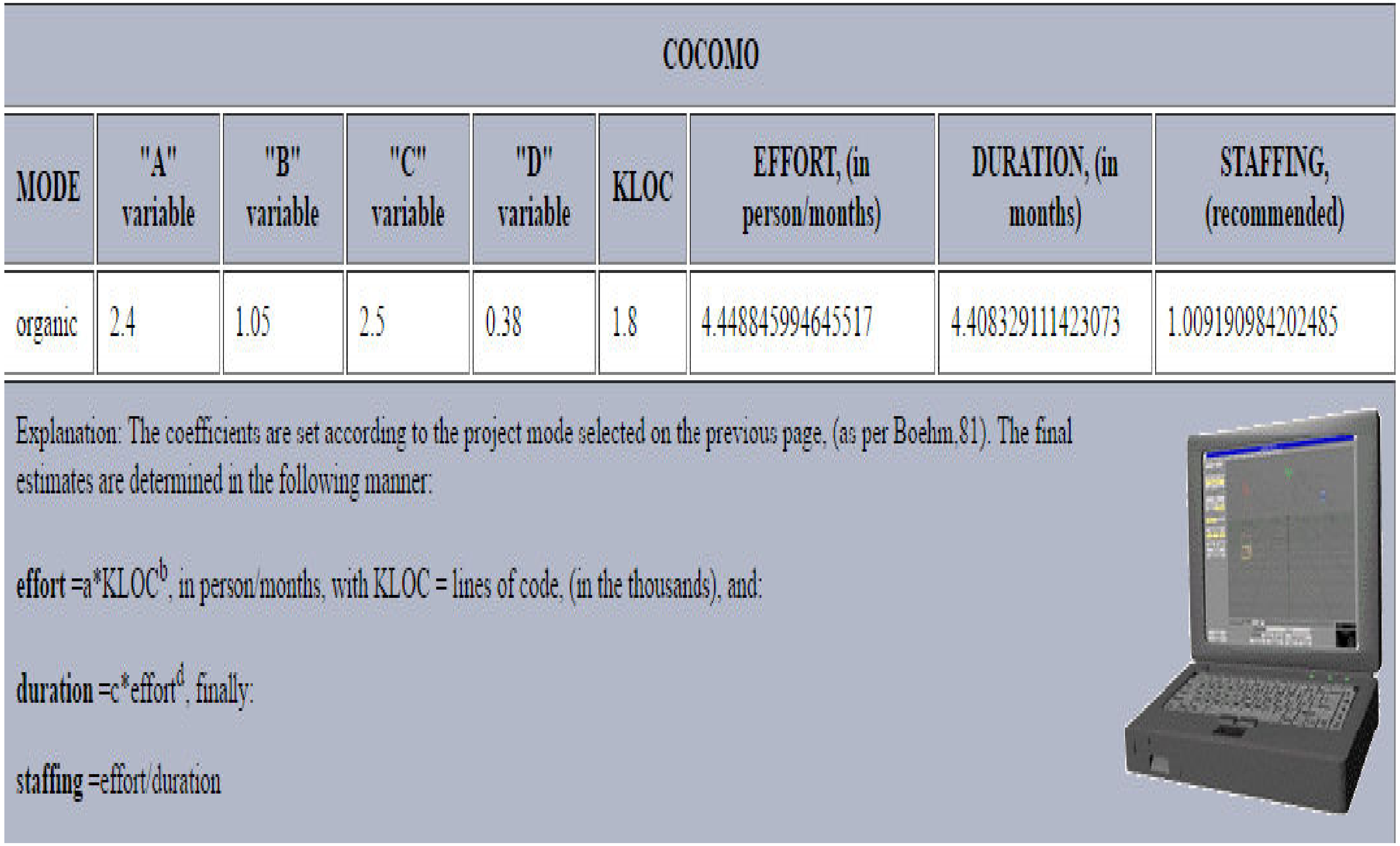
- **FRONT-END (CLIENT-SIDE):**
    - Any Windows or Linux based operating system.
    - A working internet connection.
    - An internet browser, preferably supporting H.T.M.L.5 and C.S.S.3.
- **WEB-SERVER:**
    - A Linux or UNIX environment with Apache Tomcat.
    - Although any latest web-server could host the project but Apache Tomcat is recommended for optimum performance.
- **DATABASE SERVER:**
    - The project uses MYSQL as its Database.

Hardware Interfaces:-

The minimum hardware requirements for the project to be hosted and used are:

- **CLIENT SIDE:**
  - **MINIMUM REQUIREMENTS:**
    - Intel Atom or Celeron Processor with
    - 128 Mega-Bytes of R.A.M.
    - A working internet connection.
- **SERVER SIDE:**
  - **MINIMUM REQUIREMENTS:**
    - Intel Pentium Processor with
    - 512 Mega-Bytes of R.A.M.
    - 1.5 G.B. Hard Disk for project source and web-server.
    - Storage space for Database according to database size.
  - **RECOMMENDED REQUIREMENTS:**
    - Intel Multi-Core Processor with
    - 1 Giga-Byte of R.A.M.
    - 1.5 G.B Hard Disk for project source and web-server.
    - Storage space for Database according to database size.

Constructive Cost Model:



COCOMO								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
organic	2.4	1.05	2.5	0.38	1.8	4.448845994645517	4.408329111423073	1.009190984202485

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm.81). The final estimates are determined in the following manner:

effort =a*KLOC^b, in person/months, with KLOC = lines of code, (in the thousands), and:

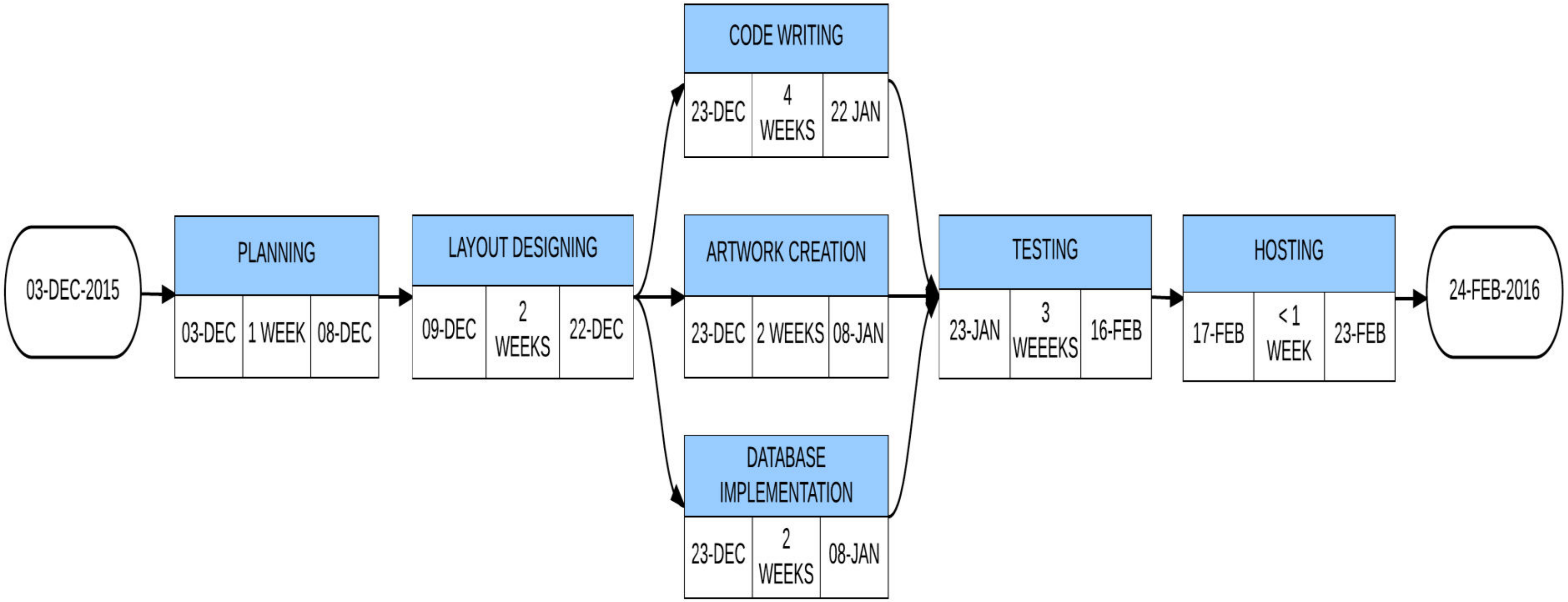
duration =c*effort^d, finally:

staffing =effort/duration

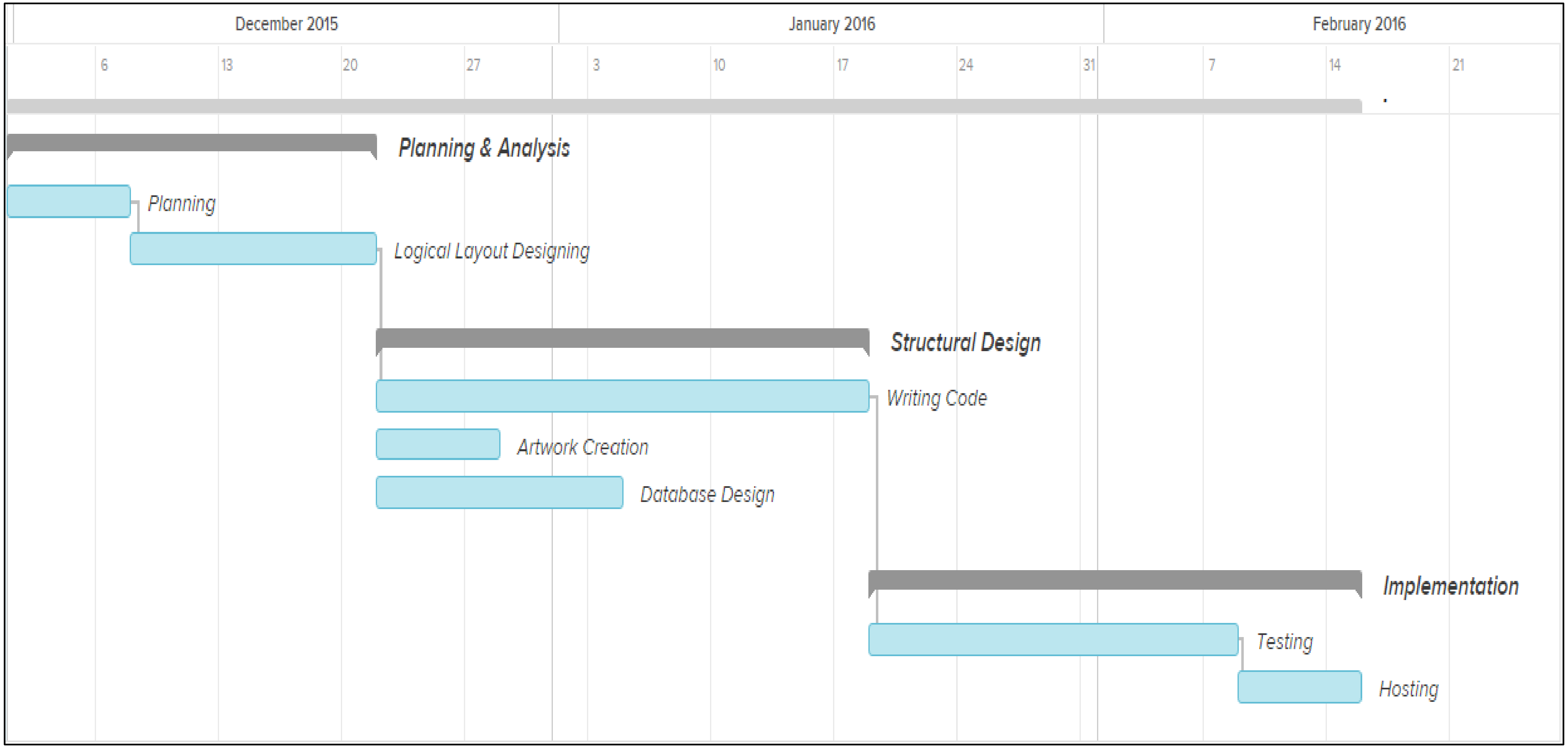
Project Scheduling:-

The project, Online Competitive Programming Portal is set to be completed within 2.8 months or around 90 days. Step by step breakout of different development processes is illustrated graphically using the following charts –

- Project Evaluation and Review Technique Chart (PERT).
- GANTT chart (Named after Henry L. Gantt).



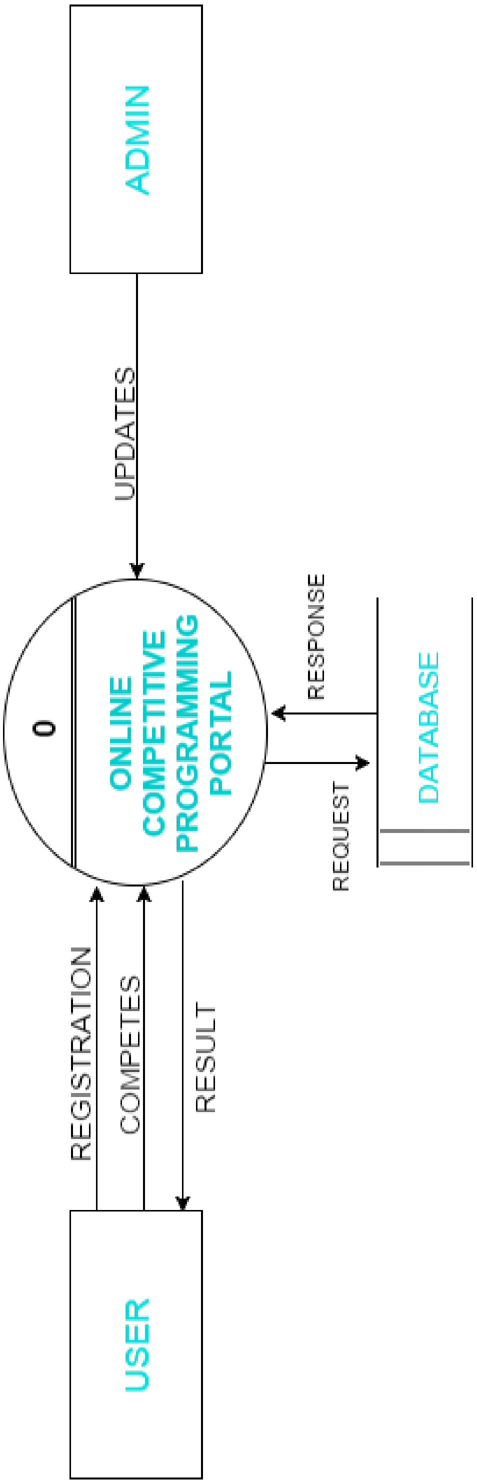
PROJECT EVALUATION AND REVIEW TECHNIQUE CHART


December 2015
January 2016
February 2016
6
13
20
27
3
10
17
24
31
7
14
21
Planning & Analysis
Planning
Logical Layout Designing
Structural Design
Writing Code
Artwork Creation
Database Design
Implementation
Testing
Hosting

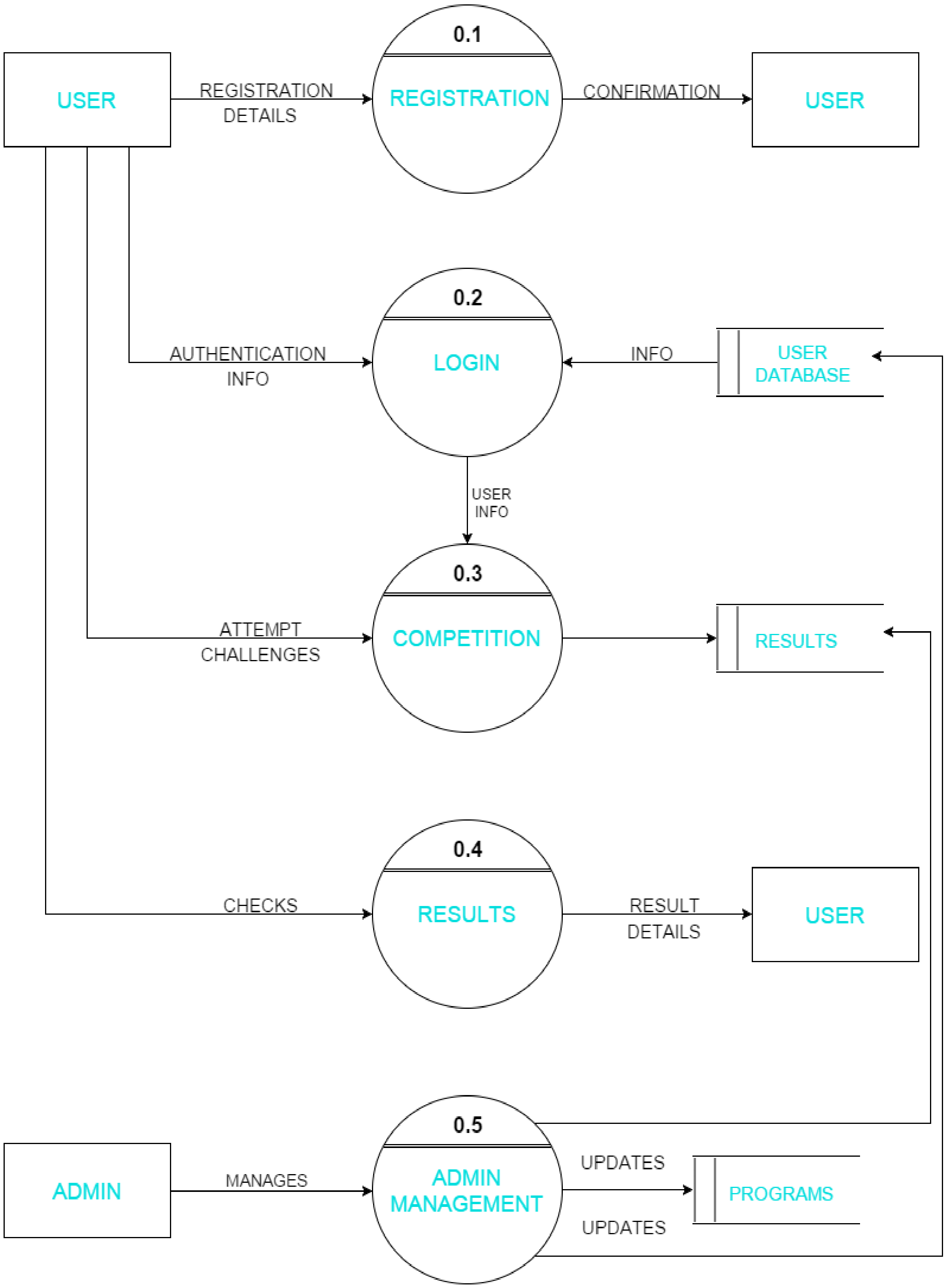
GANTT CHART

Data-Flow:-

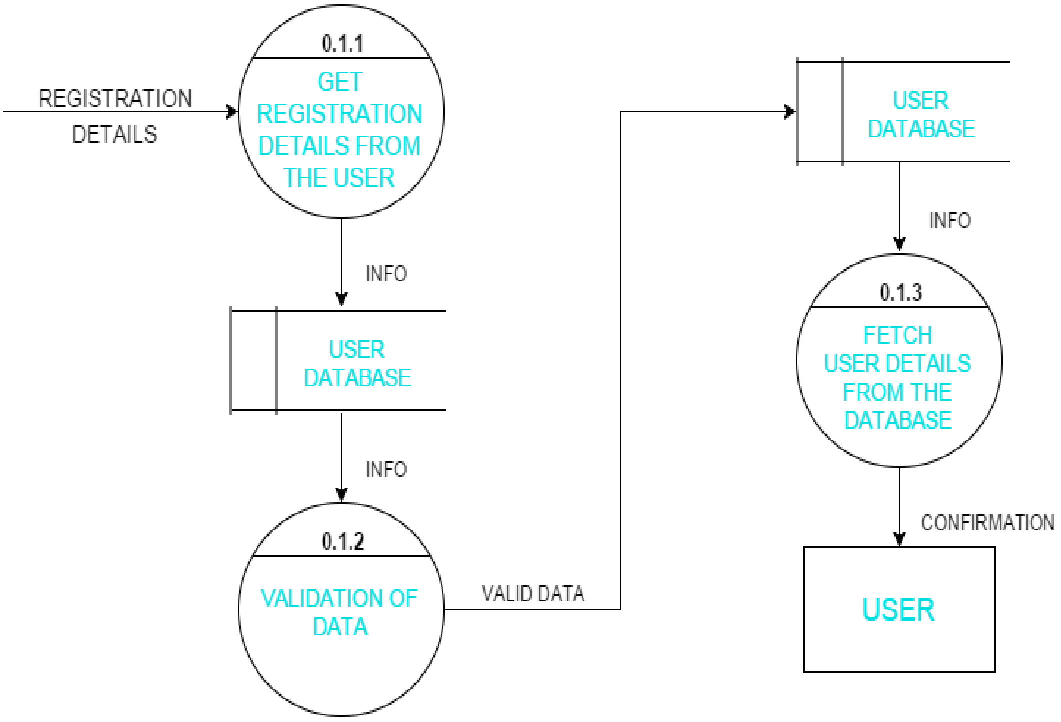
The flow of data inside the project could be represented as:



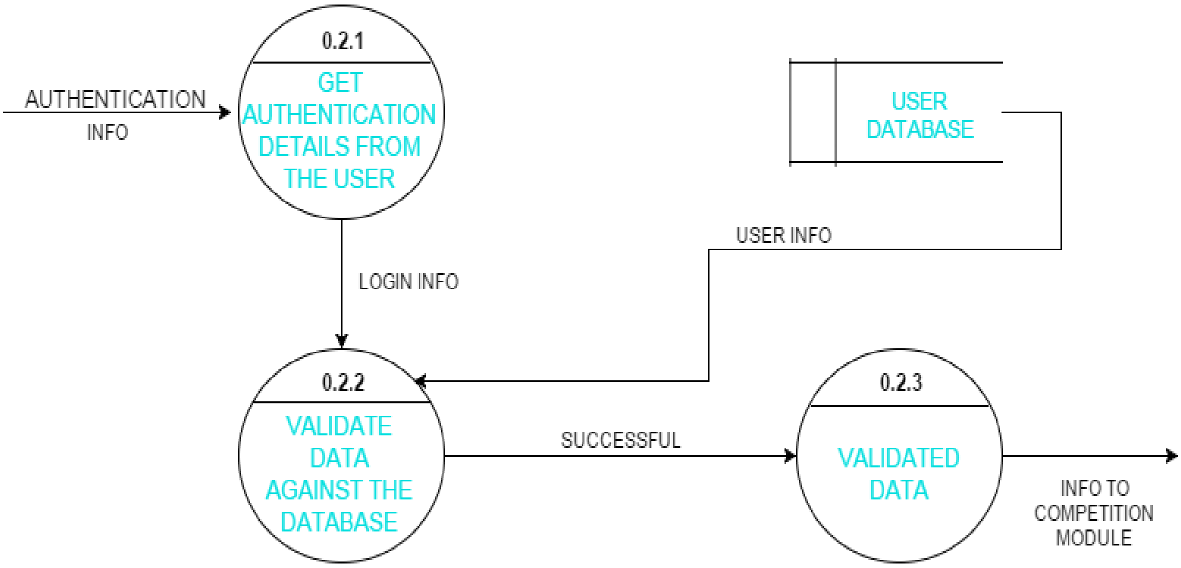
LEVEL- 0
CONTEXT LEVEL
DATA FLOW DIAGRAM



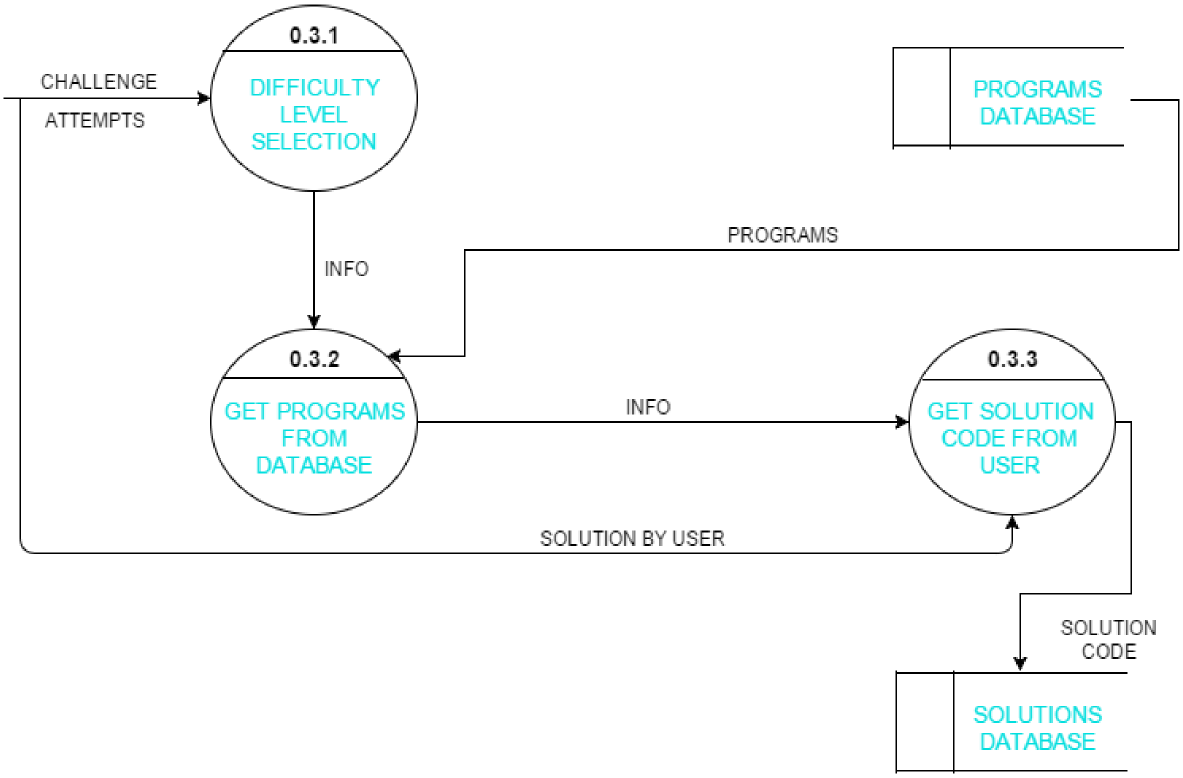
LEVEL- 1
DATA FLOW DIAGRAM



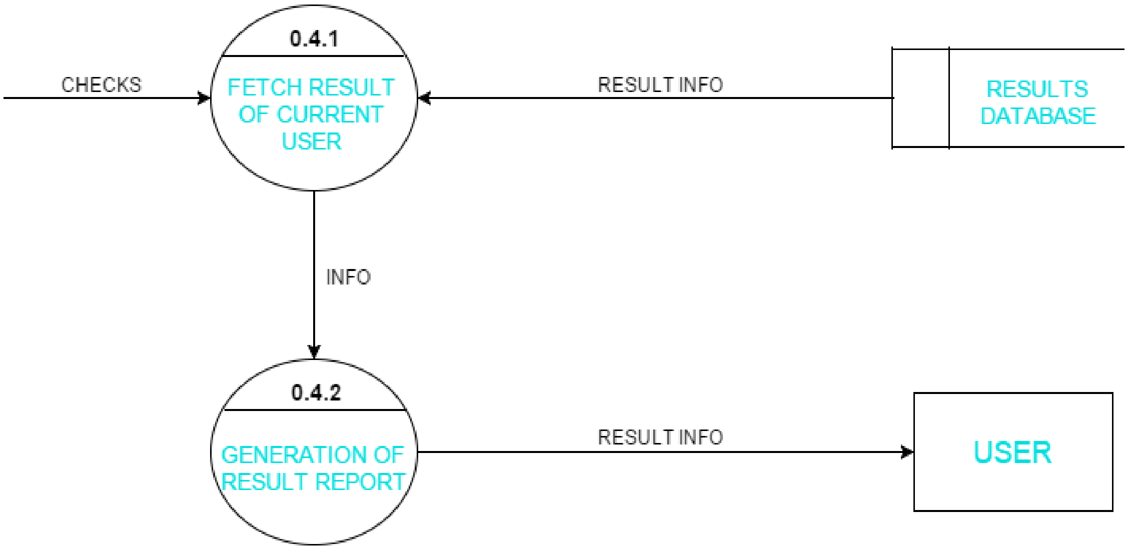
LEVEL - 2
REGISTRATION MODULE
DATA FLOW DIAGRAM


0.2.1
GET AUTHENTICATION DETAILS FROM THE USER
AUTHENTICATION INFO
USER DATABASE
USER INFO
LOGIN INFO
0.2.2
VALIDATE DATA AGAINST THE DATABASE
SUCCESSFUL
0.2.3
VALIDATED DATA
INFO TO COMPETITION MODULE

LEVEL - 2
LOGIN MODULE
DATA FLOW DIAGRAM

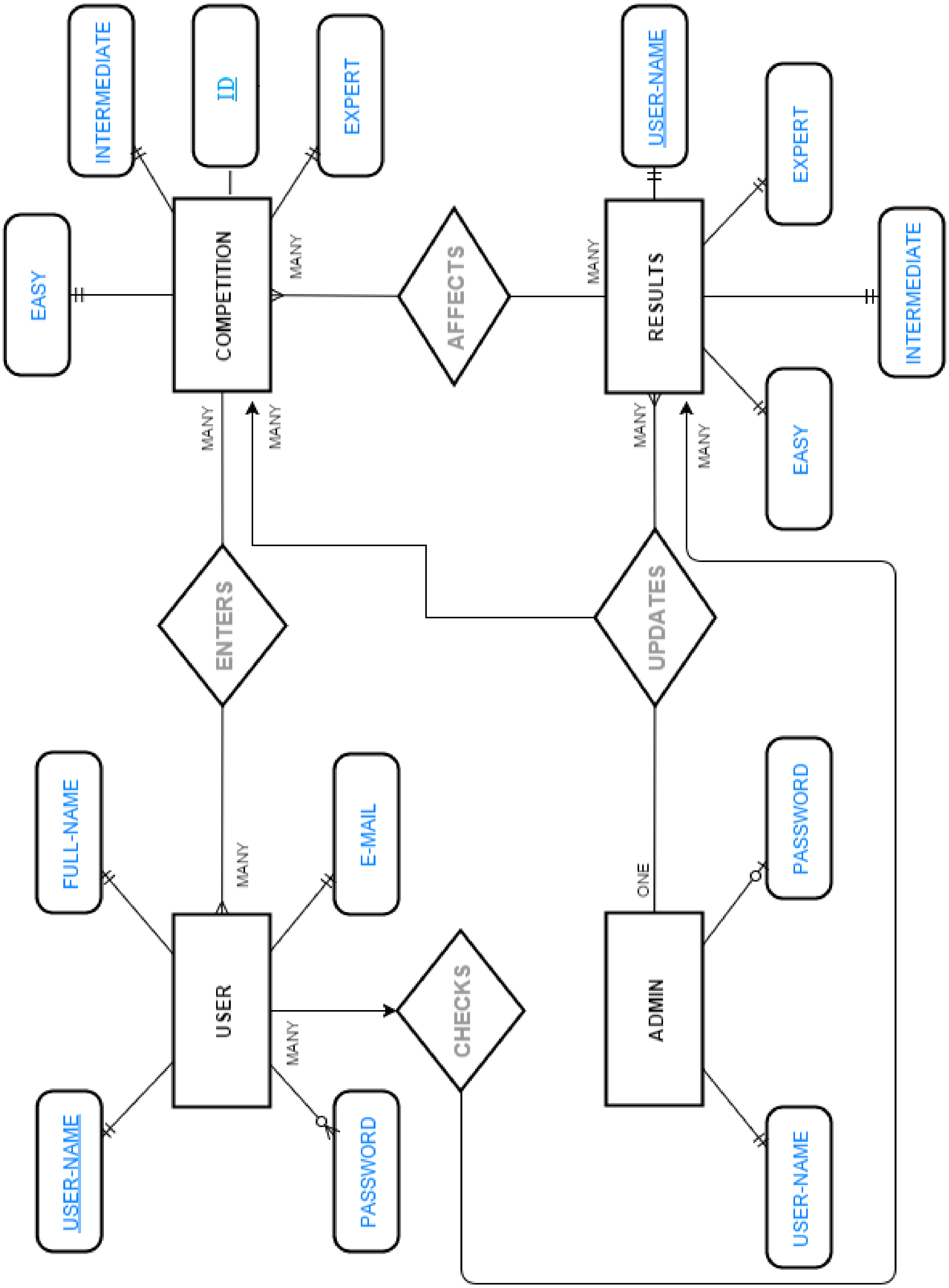


LEVEL - 2
COMPETITION MODULE
DATA FLOW DIAGRAM

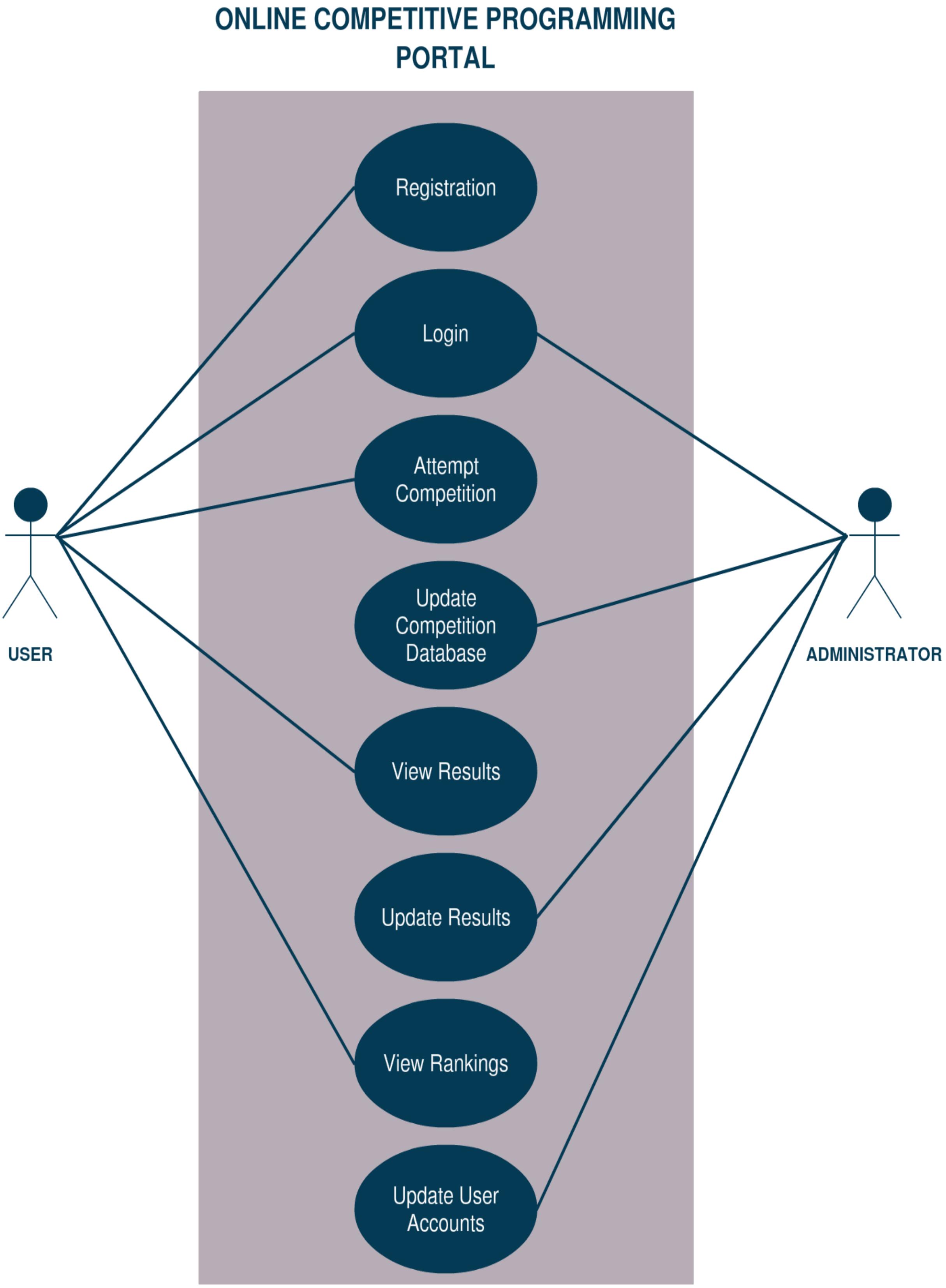


LEVEL - 2
RESULTS MODULE
DATA FLOW DIAGRAM

Entity Relationship Diagram:-

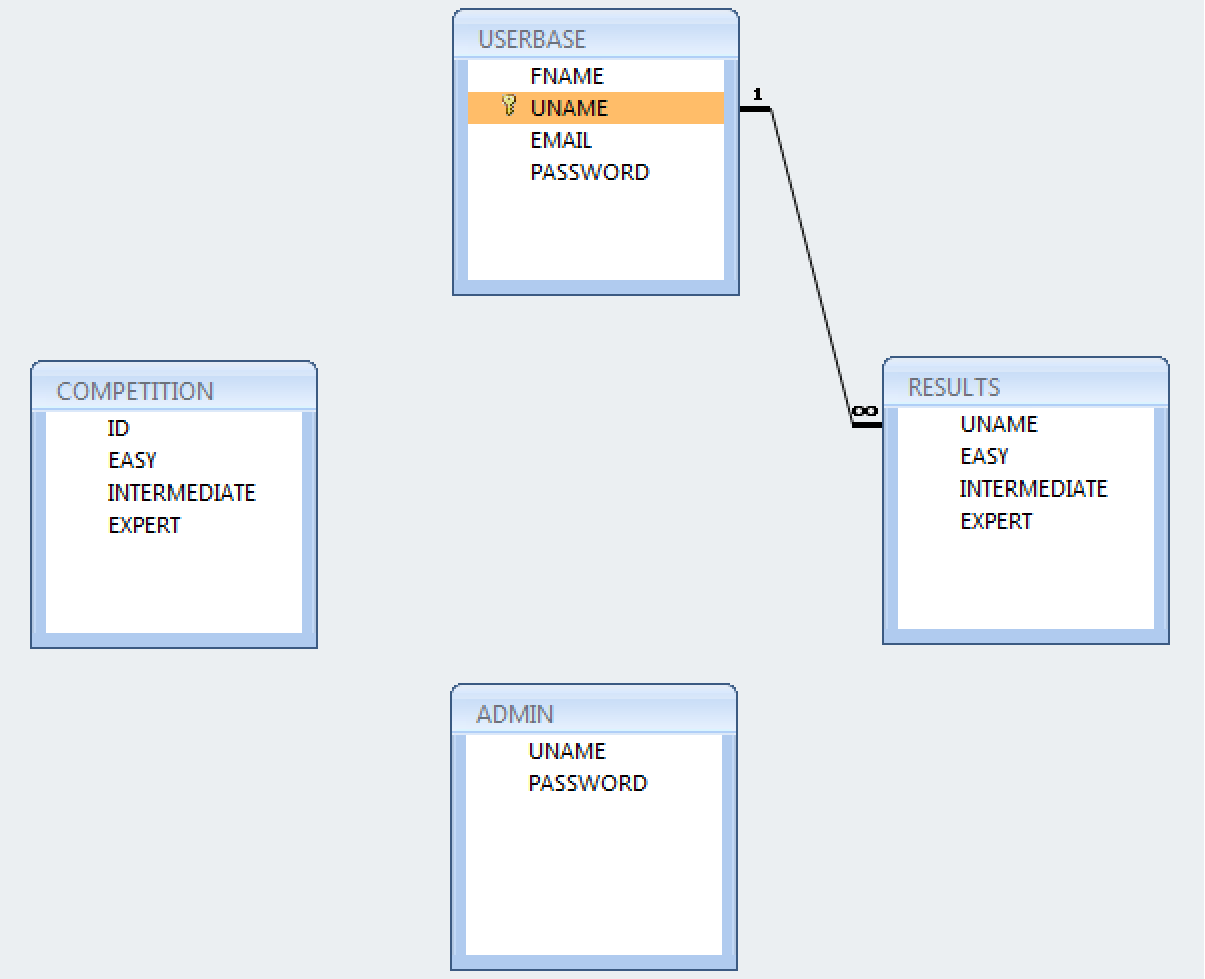


Use Case Diagram:-



DESIGN DOCUMENT

Database Design:-



Data Dictionary:-

USERBASE :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	FNAME	VARCHAR-2	4000 bytes		Full Name Of The User
2	UNAME	VARCHAR-2	4000 bytes	P.K.	User Name Of The User
3	EMAIL	VARCHAR-2	4000 bytes		E-mail Of The User
4	PASSWORD	VARCHAR-2	4000 bytes		Password Of The User

COMPETITION :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	ID	AUTO	4 bytes	P.K.	I.D. Of The Question
2	EASY	VARCHAR-2	4000 bytes		Easy Questions
3	INTERMEDIATE	VARCHAR-2	4000 bytes		Mediate Level Questions
4	EXPERT	VARCHAR-2	4000 bytes		Expert Level Questions

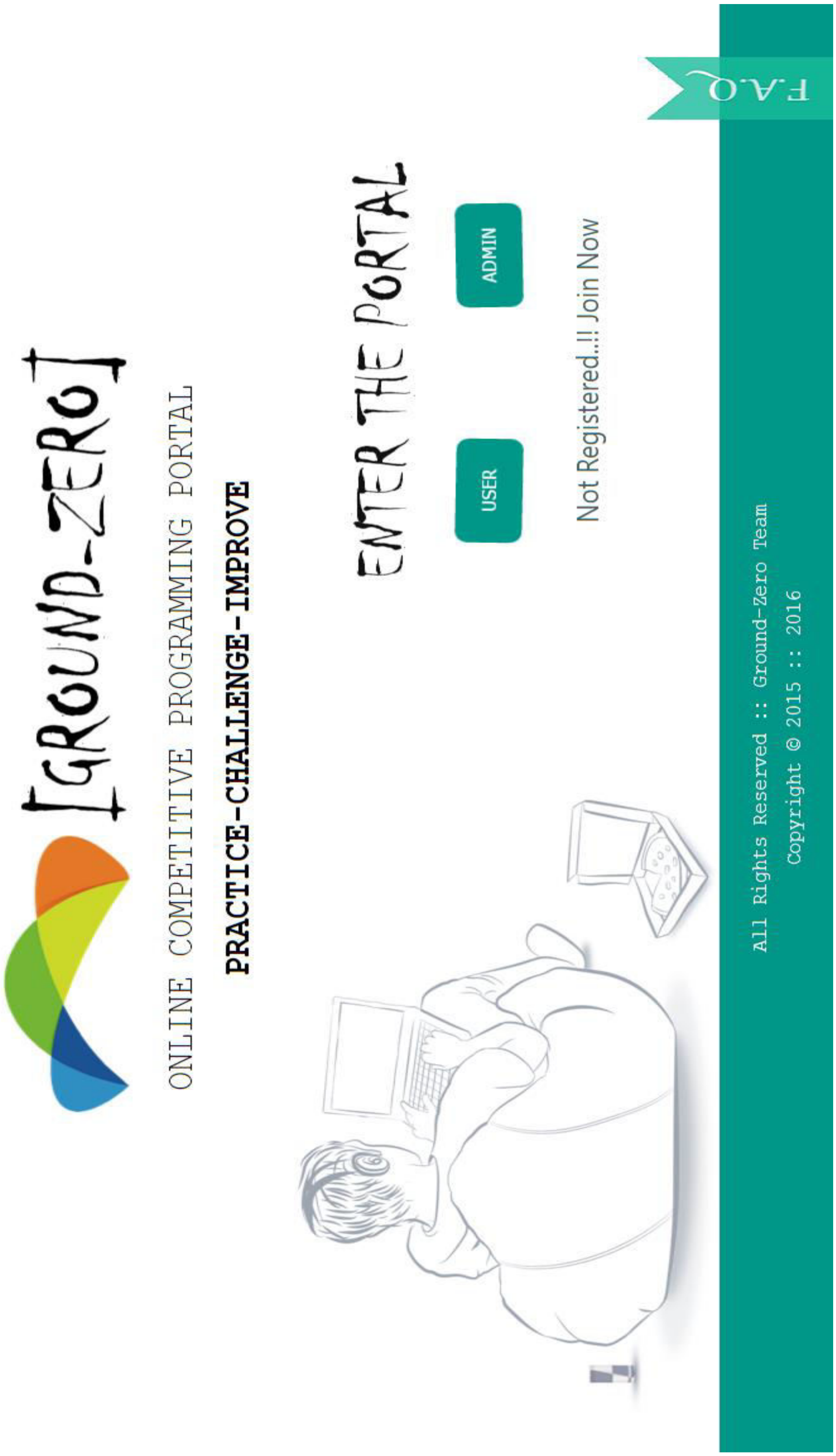
RESULTS :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	UNAME	VARCHAR-2	4000 bytes	F.K.	User Name Of The User
2	EASY	VARCHAR-2	4000 bytes		Easy Questions
3	INTERMEDIATE	VARCHAR-2	4000 bytes		Mediate Level Questions
4	EXPERT	VARCHAR-2	4000 bytes		Expert Level Questions

ADMIN :-

ID	FIELD NAME	DATA TYPE	SIZE (MAX)	CONSTRAINT	DESCRIPTION
1	UNAME	VARCHAR-2	4000 bytes		User Name Of The Admin
2	PASSWORD	VARCHAR-2	4000 bytes		Password Of The Admin

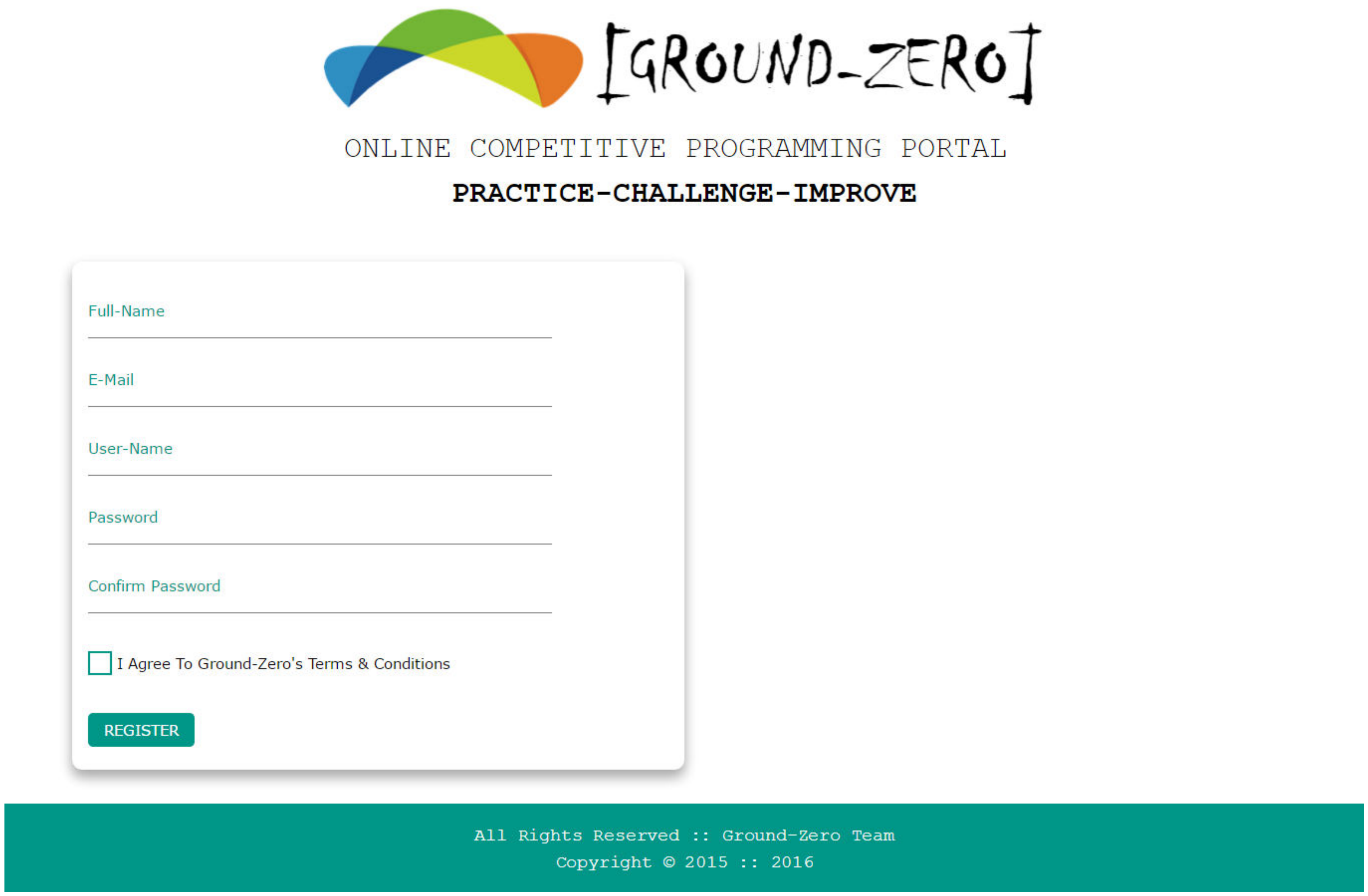
SCREENSHOT


[GROUND-ZERO]
ONLINE COMPETITIVE PROGRAMMING PORTAL
PRACTICE-CHALLENGE-IMPROVE
ENTER THE PORTAL
USER
ADMIN
F.A.Q

Not Registered..!! Join Now

All Rights Reserved :: Ground-Zero Team

Copyright © 2015 :: 2016



[GROUND-ZERO]

ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

Full-Name

E-Mail

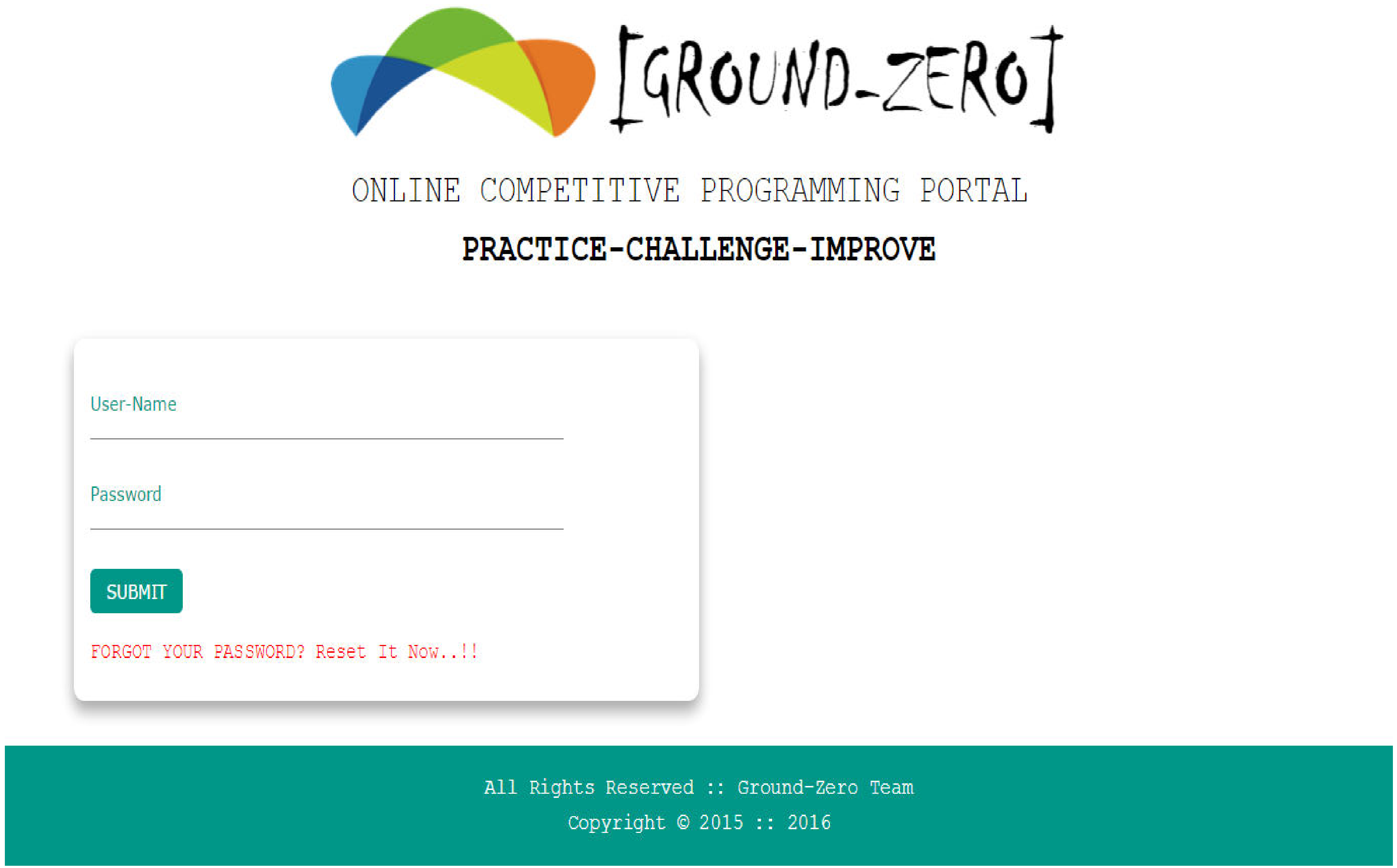
User-Name

Password

Confirm Password

□ I Agree To Ground-Zero's Terms & Conditions

All Rights Reserved :: Ground-Zero Team
Copyright © 2015 :: 2016



[GROUND-ZERO]

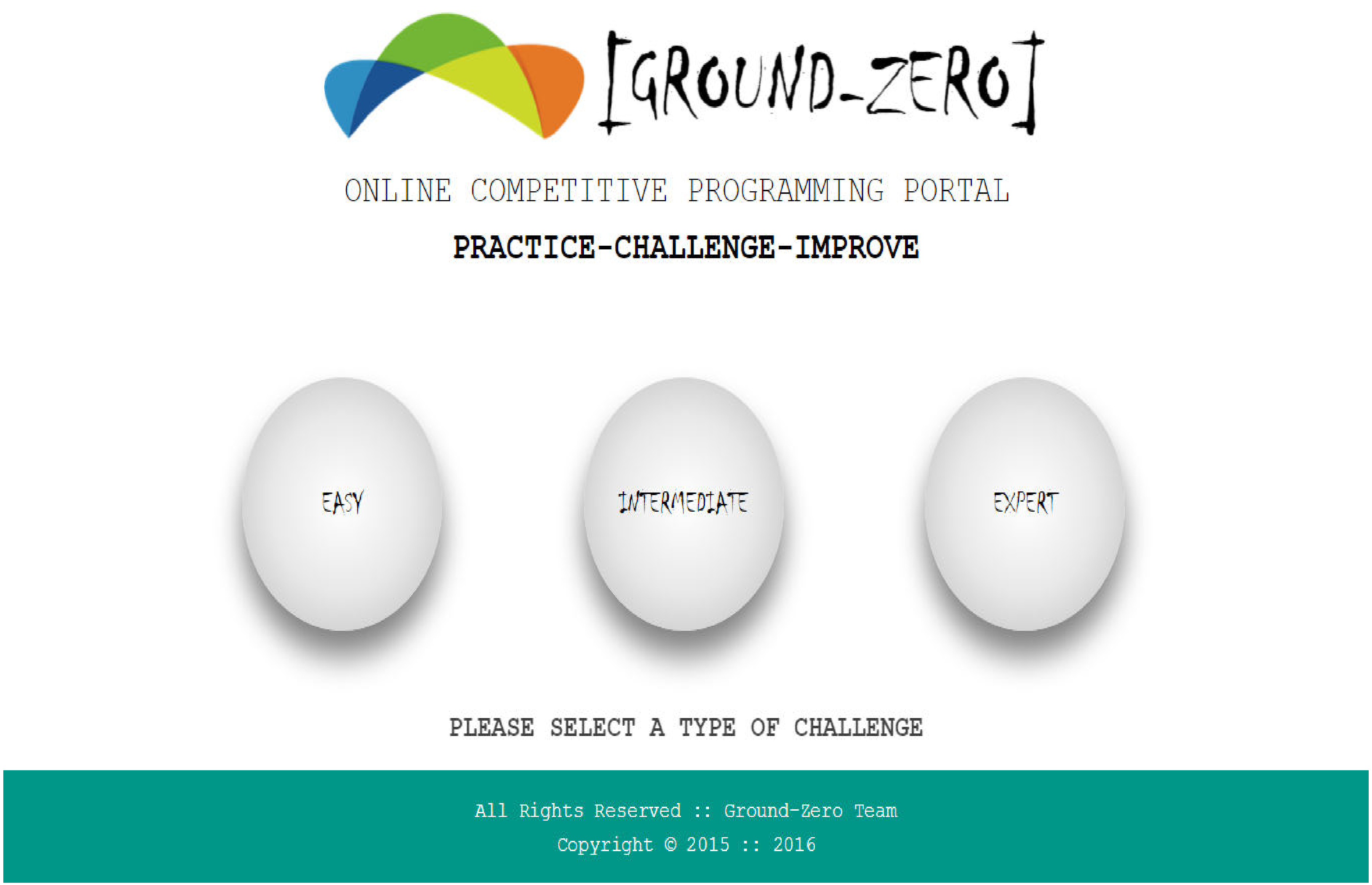
ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

FORGOT YOUR PASSWORD? Reset It Now..!!

All Rights Reserved :: Ground-Zero Team

Copyright © 2015 :: 2016



[GROUND-ZERO]

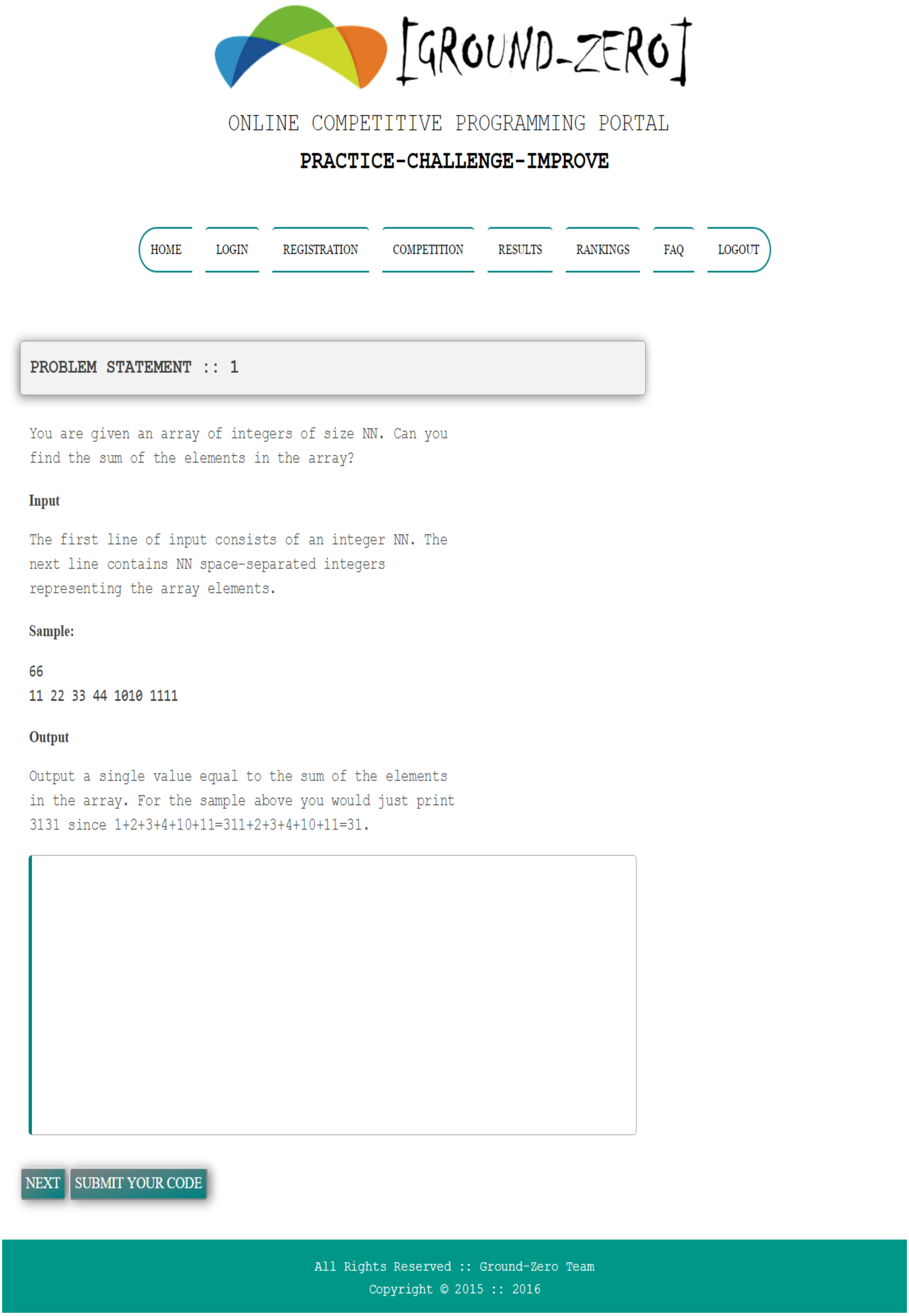
ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

PLEASE SELECT A TYPE OF CHALLENGE

All Rights Reserved :: Ground-Zero Team

Copyright © 2015 :: 2016



ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

PROBLEM STATEMENT :: 1

You are given an array of integers of size NN. Can you find the sum of the elements in the array?

Input

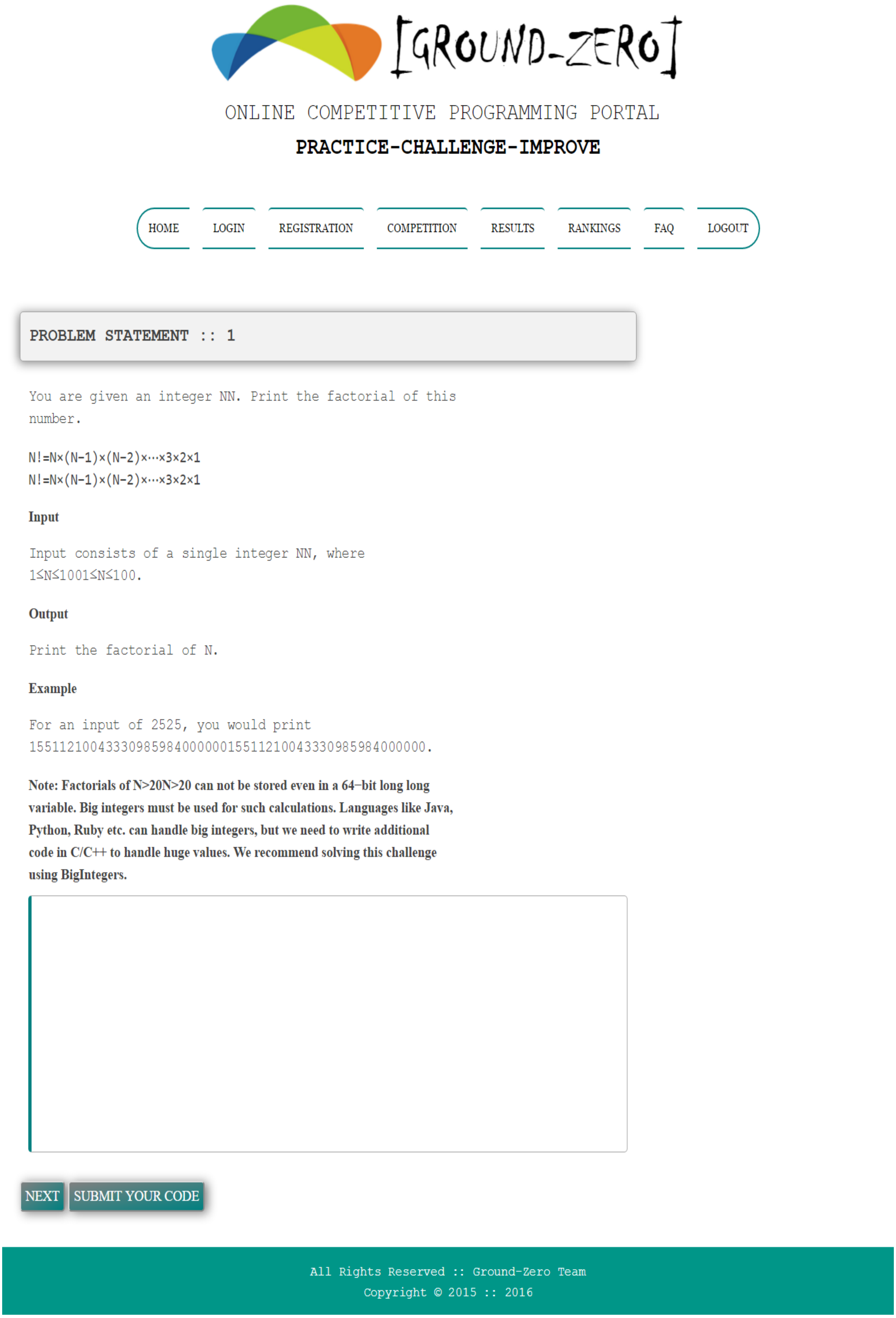
The first line of input consists of an integer NN. The next line contains NN space-separated integers representing the array elements.

Sample:

```
66
11 22 33 44 1010 1111
```

Output

Output a single value equal to the sum of the elements in the array. For the sample above you would just print 3131 since 1+2+3+4+10+11=311+2+3+4+10+11=31.



ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

PROBLEM STATEMENT :: 1

You are given an integer NN. Print the factorial of this number.

N!=N×(N−1)×(N−2)×⋯×3×2×1
N!=N×(N−1)×(N−2)×⋯×3×2×1

Input

Input consists of a single integer NN, where 1≤N≤1001≤N≤100.

Output

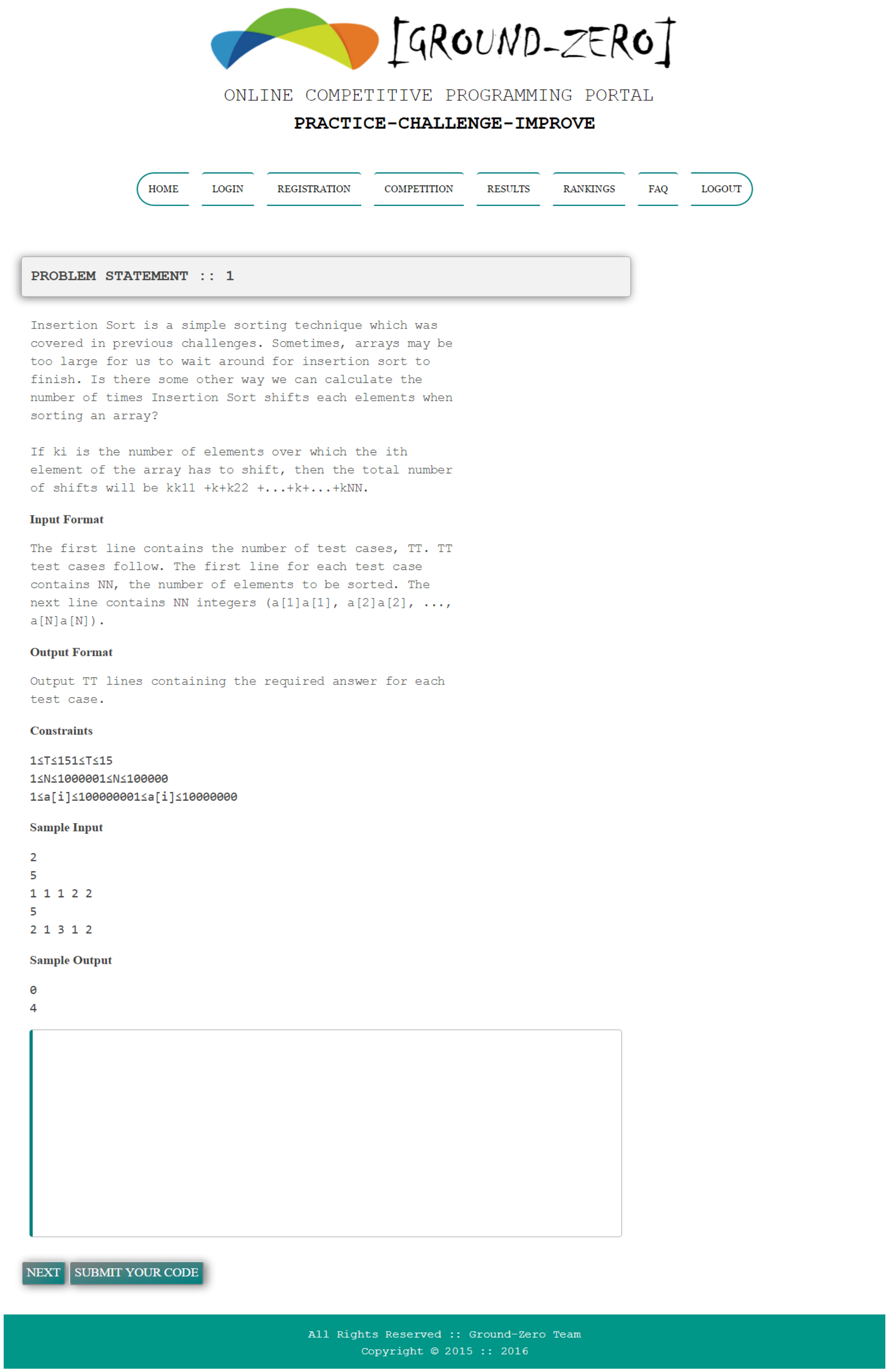
Print the factorial of N.

Example

For an input of 2525, you would print 1551121004333098598400000015511210043330985984000000.

Note: Factorials of N>20N>20 can not be stored even in a 64−bit long long variable. Big integers must be used for such calculations. Languages like Java, Python, Ruby etc. can handle big integers, but we need to write additional code in C/C++ to handle huge values. We recommend solving this challenge using BigIntegers.

All Rights Reserved :: Ground-Zero Team
Copyright © 2015 :: 2016



PROBLEM STATEMENT :: 1

Insertion Sort is a simple sorting technique which was covered in previous challenges. Sometimes, arrays may be too large for us to wait around for insertion sort to finish. Is there some other way we can calculate the number of times Insertion Sort shifts each elements when sorting an array?

If ki is the number of elements over which the ith element of the array has to shift, then the total number of shifts will be kk11 +k+k22 +...+k+...+kNN.

Input Format

The first line contains the number of test cases, TT. TT test cases follow. The first line for each test case contains NN, the number of elements to be sorted. The next line contains NN integers (a[1]a[1], a[2]a[2], ..., a[N]a[N]).

Output Format

Output TT lines containing the required answer for each test case.

Constraints

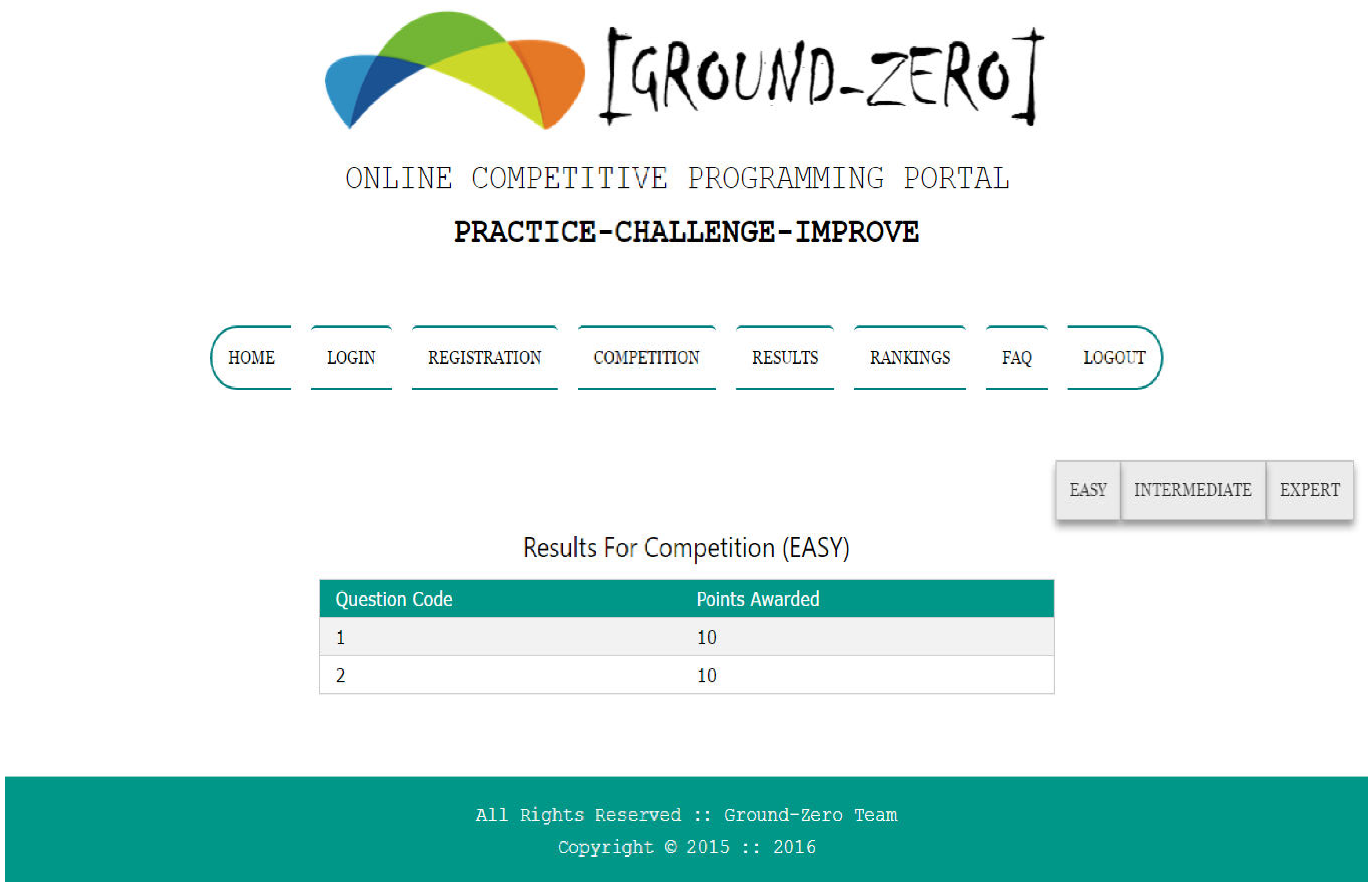
1≤T≤151≤T≤15
1≤N≤1000001≤N≤100000
1≤a[i]≤100000001≤a[i]≤10000000

Sample Input

```
2
5
1 1 1 2 2
5
2 1 3 1 2
```

Sample Output

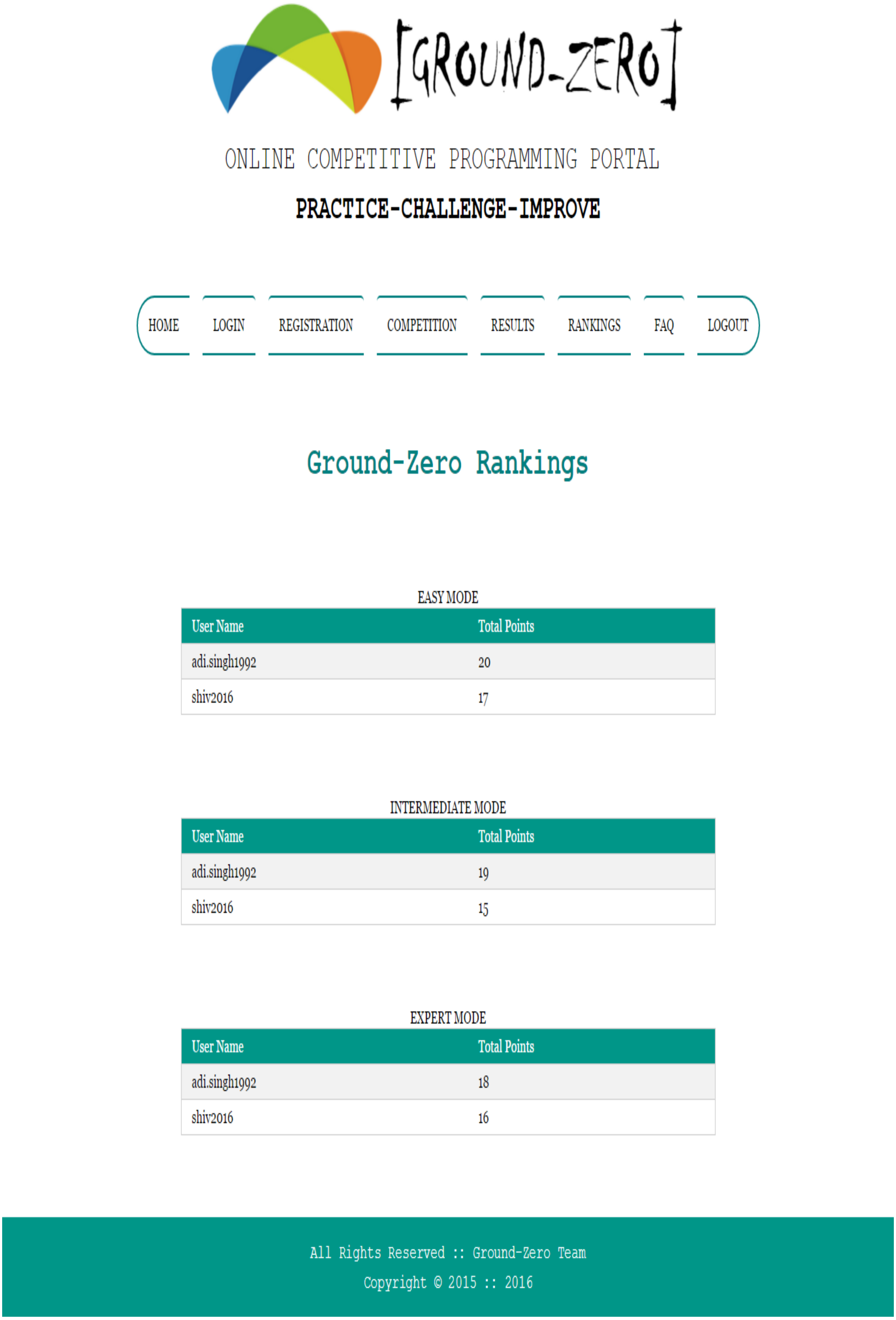
```
0
4
```



Results For Competition (EASY)

Question Code	Points Awarded
1	10
2	10

All Rights Reserved :: Ground-Zero Team
Copyright © 2015 :: 2016



ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

Ground-Zero Rankings

EASY MODE

User Name	Total Points
adi.singh1992	20
shiv2016	17

INTERMEDIATE MODE

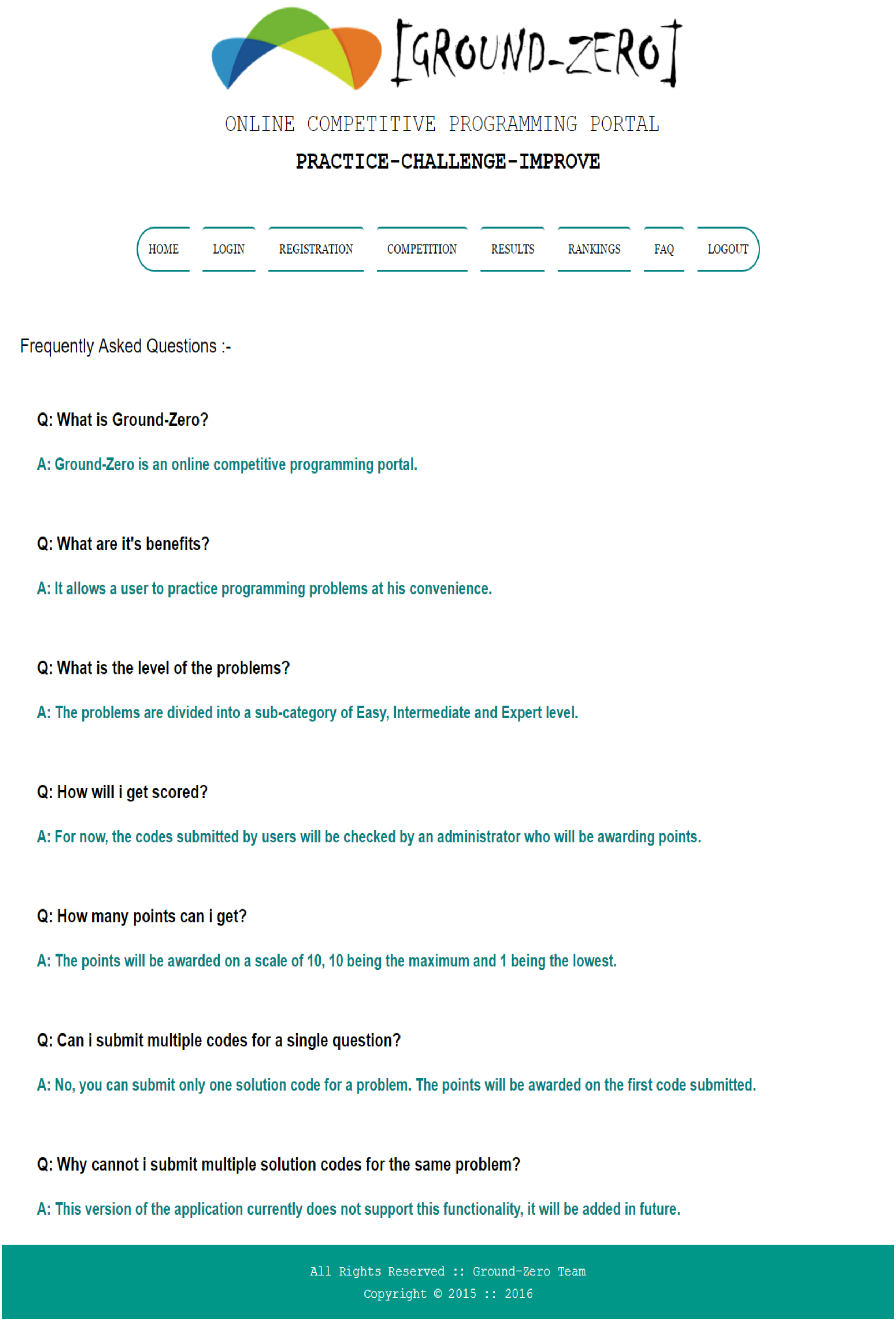
User Name	Total Points
adi.singh1992	19
shiv2016	15

EXPERT MODE

User Name	Total Points
adi.singh1992	18
shiv2016	16

All Rights Reserved :: Ground-Zero Team

Copyright © 2015 :: 2016



ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

Frequently Asked Questions :-

Q: What is Ground-Zero?

A: Ground-Zero is an online competitive programming portal.

Q: What are it's benefits?

A: It allows a user to practice programming problems at his convenience.

Q: What is the level of the problems?

A: The problems are divided into a sub-category of Easy, Intermediate and Expert level.

Q: How will i get scored?

A: For now, the codes submitted by users will be checked by an administrator who will be awarding points.

Q: How many points can i get?

A: The points will be awarded on a scale of 10, 10 being the maximum and 1 being the lowest.

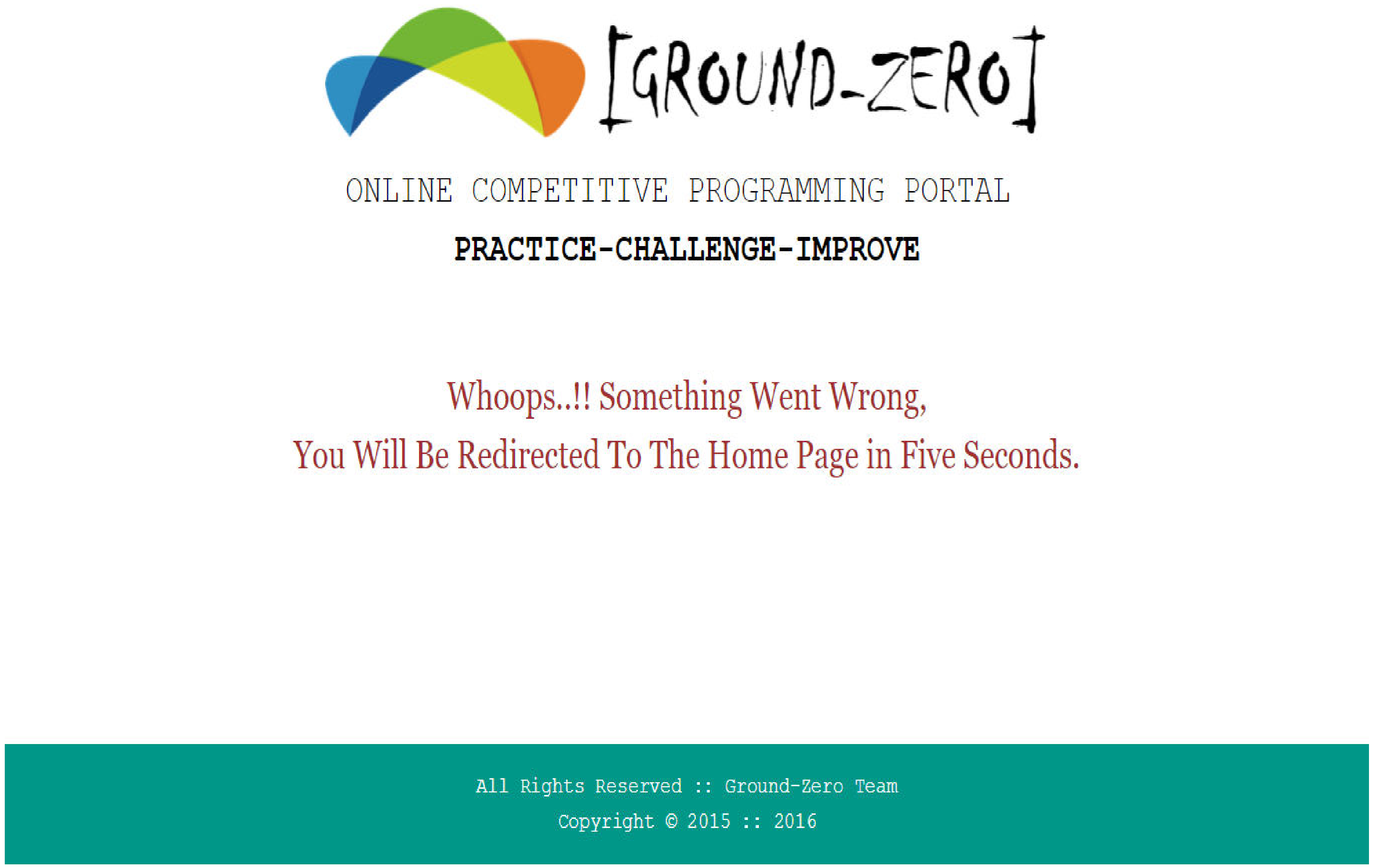
Q: Can i submit multiple codes for a single question?

A: No, you can submit only one solution code for a problem. The points will be awarded on the first code submitted.

Q: Why cannot i submit multiple solution codes for the same problem?

A: This version of the application currently does not support this functionality, it will be added in future.

All Rights Reserved :: Ground-Zero Team
Copyright © 2015 :: 2016



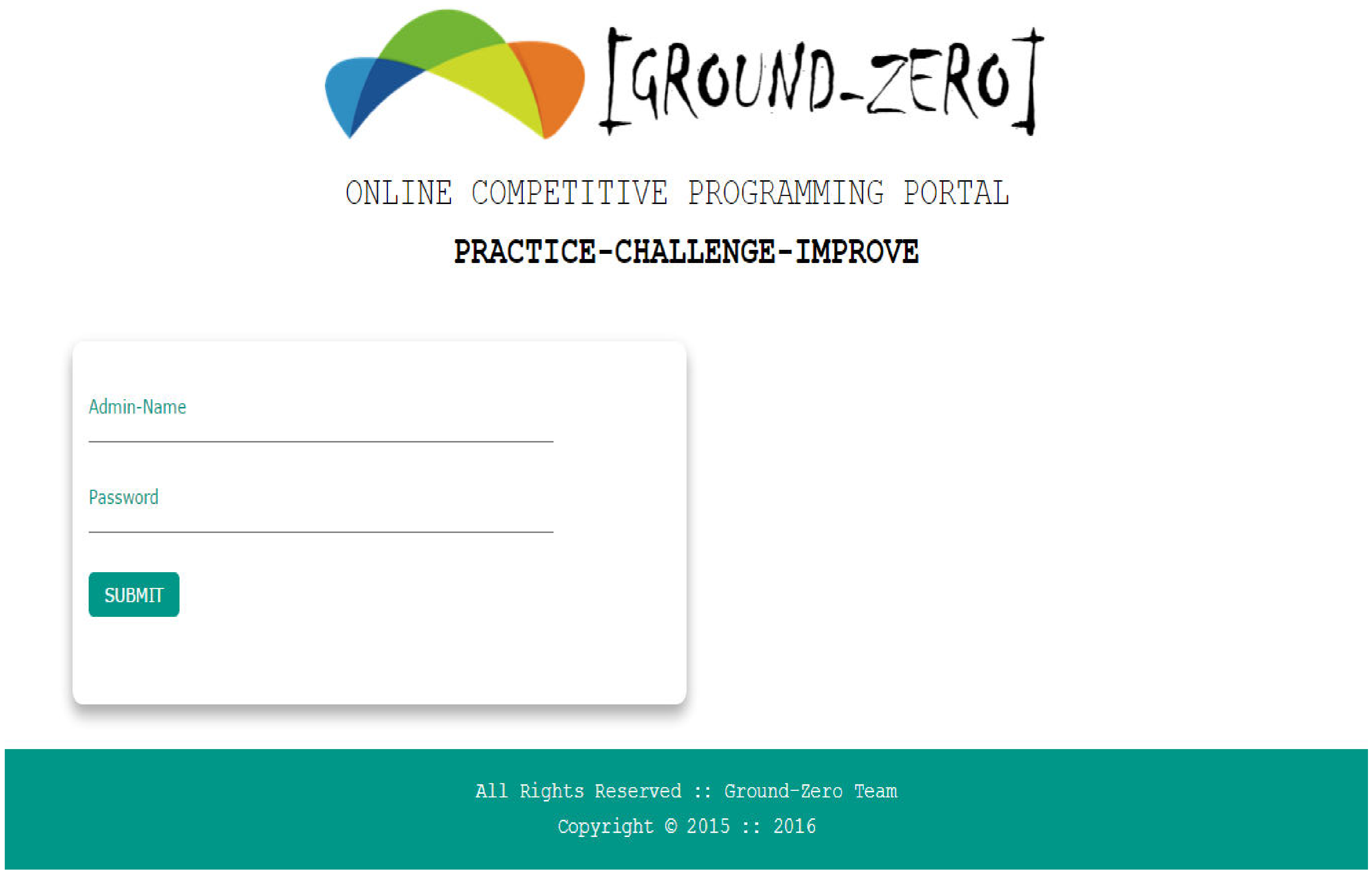
[GROUND-ZERO]

ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

Whoops..!! Something Went Wrong,
You Will Be Redirected To The Home Page in Five Seconds.

All Rights Reserved :: Ground-Zero Team
Copyright © 2015 :: 2016

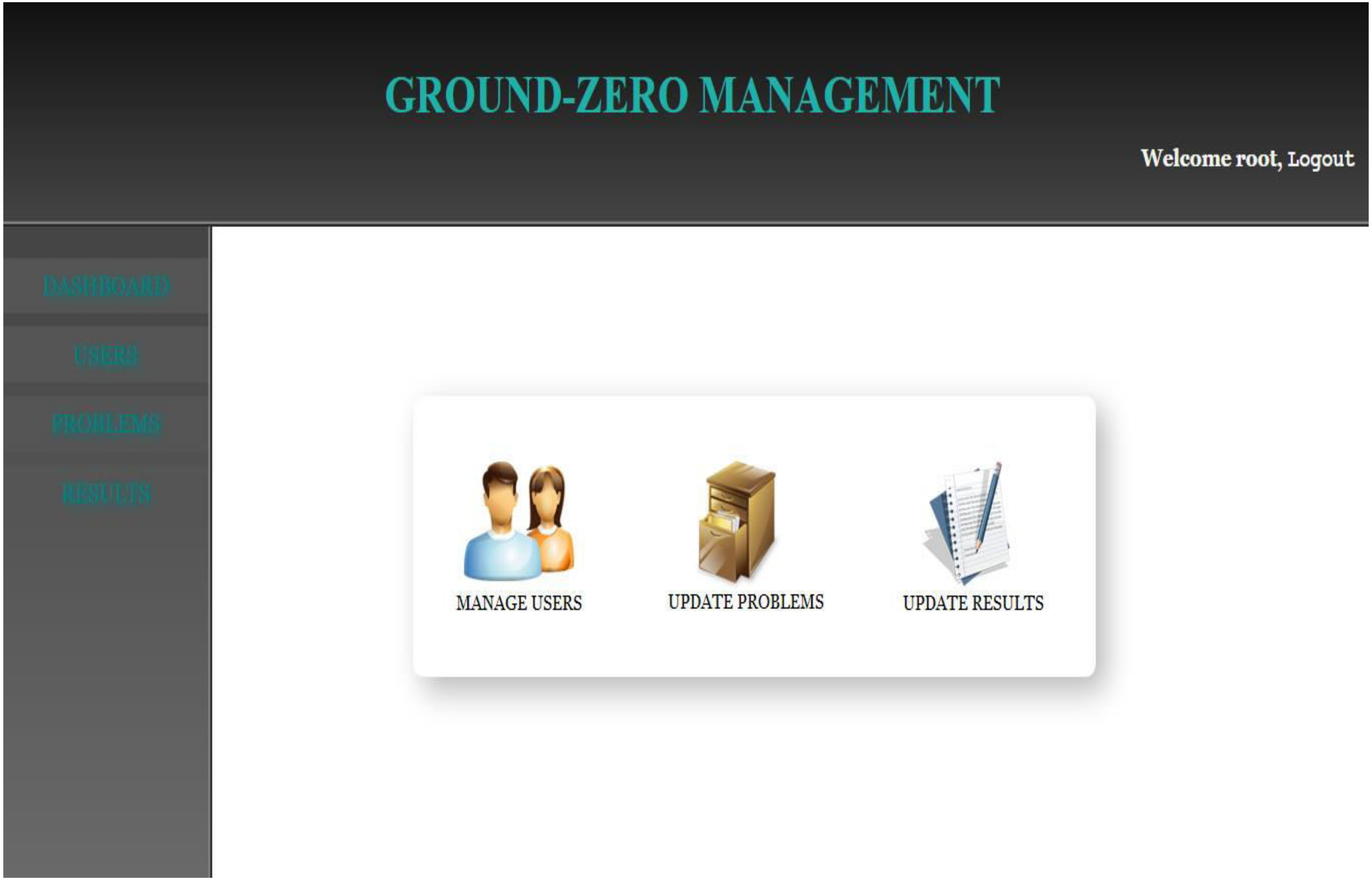


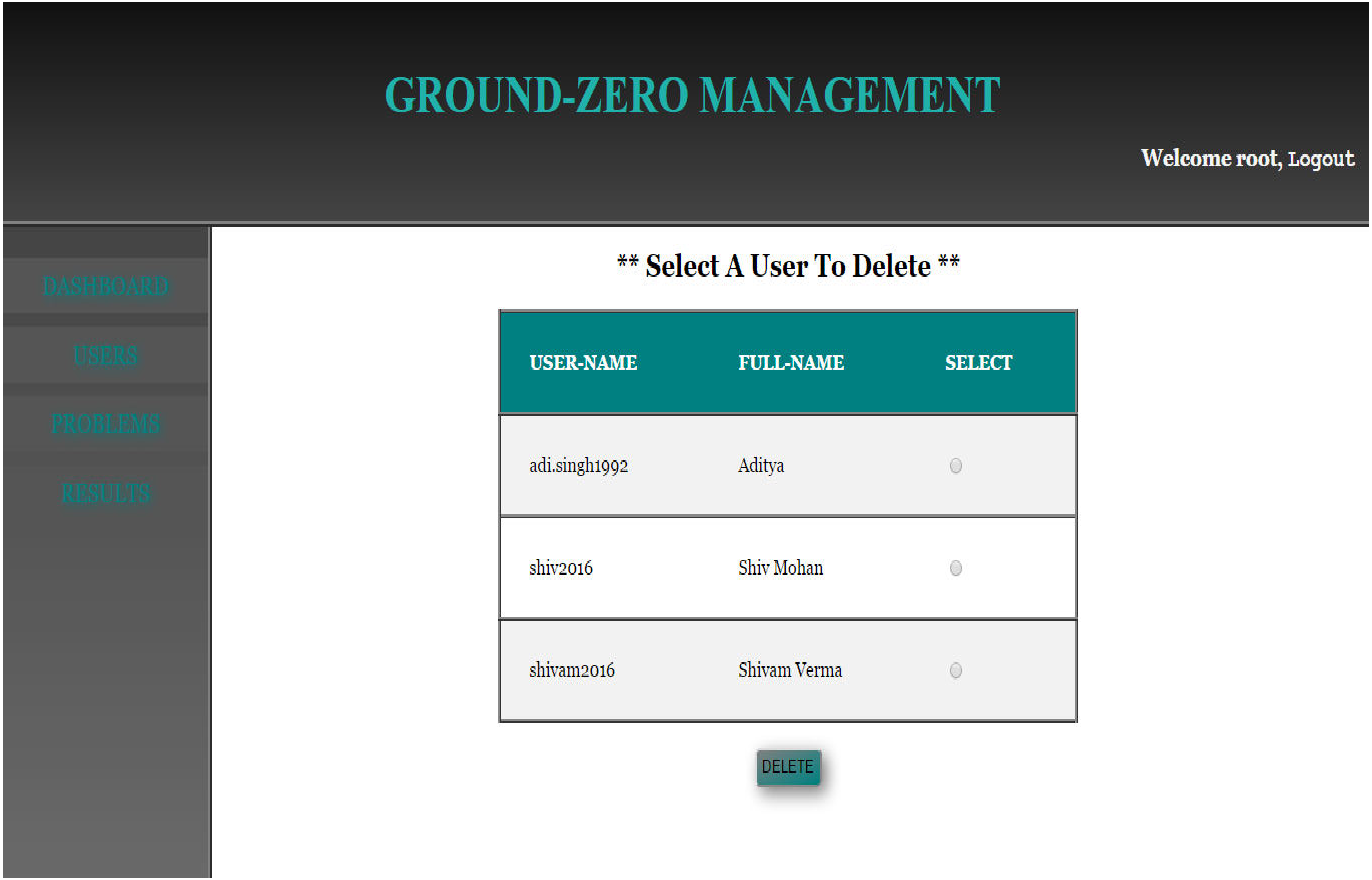
ONLINE COMPETITIVE PROGRAMMING PORTAL

PRACTICE-CHALLENGE-IMPROVE

All Rights Reserved :: Ground-Zero Team

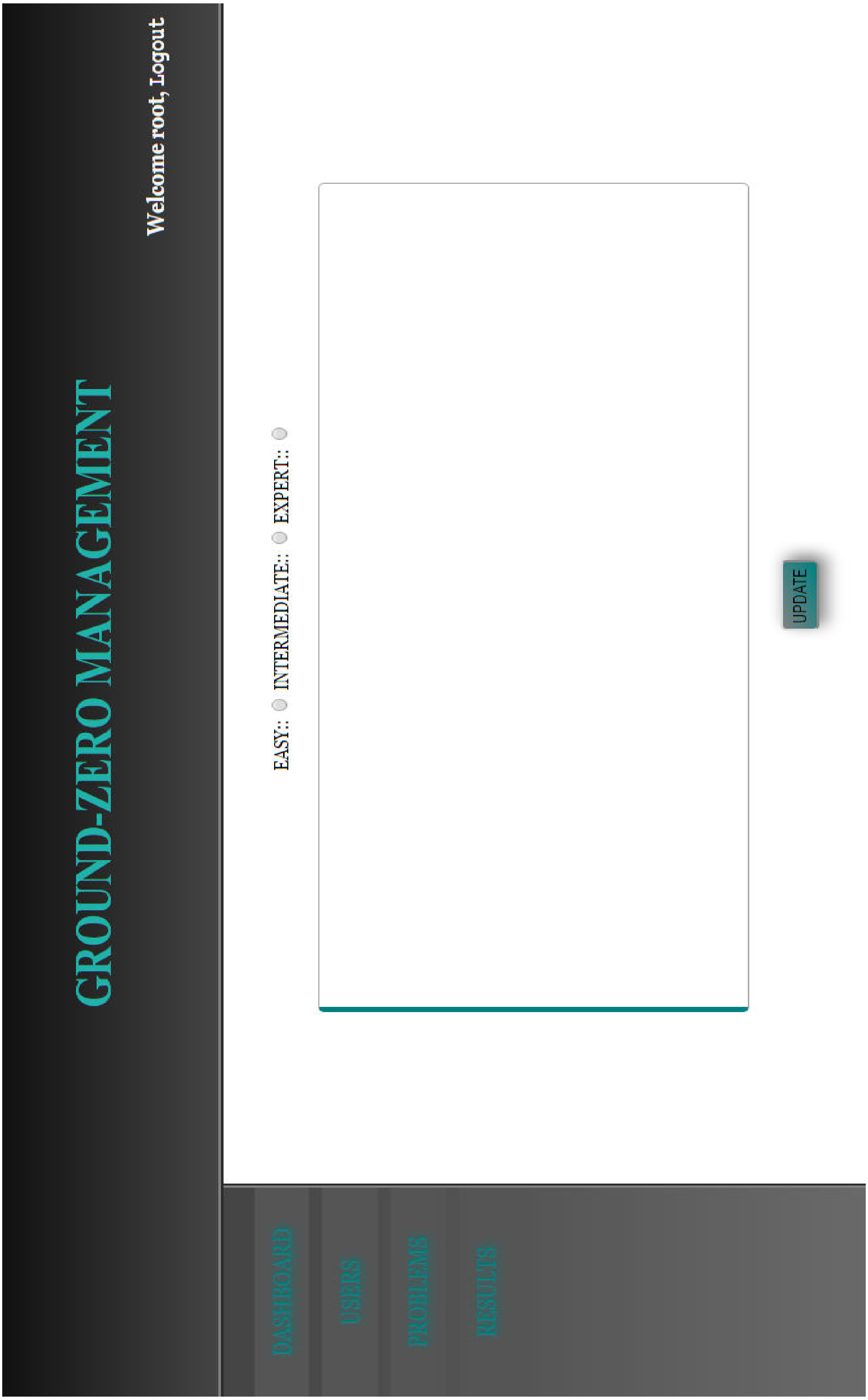
Copyright © 2015 :: 2016


GROUND-ZERO MANAGEMENT
Welcome root, Logout
DASHBOARD
USERS
PROBLEMS
RESULTS
MANAGE USERS
UPDATE PROBLEMS
UPDATE RESULTS

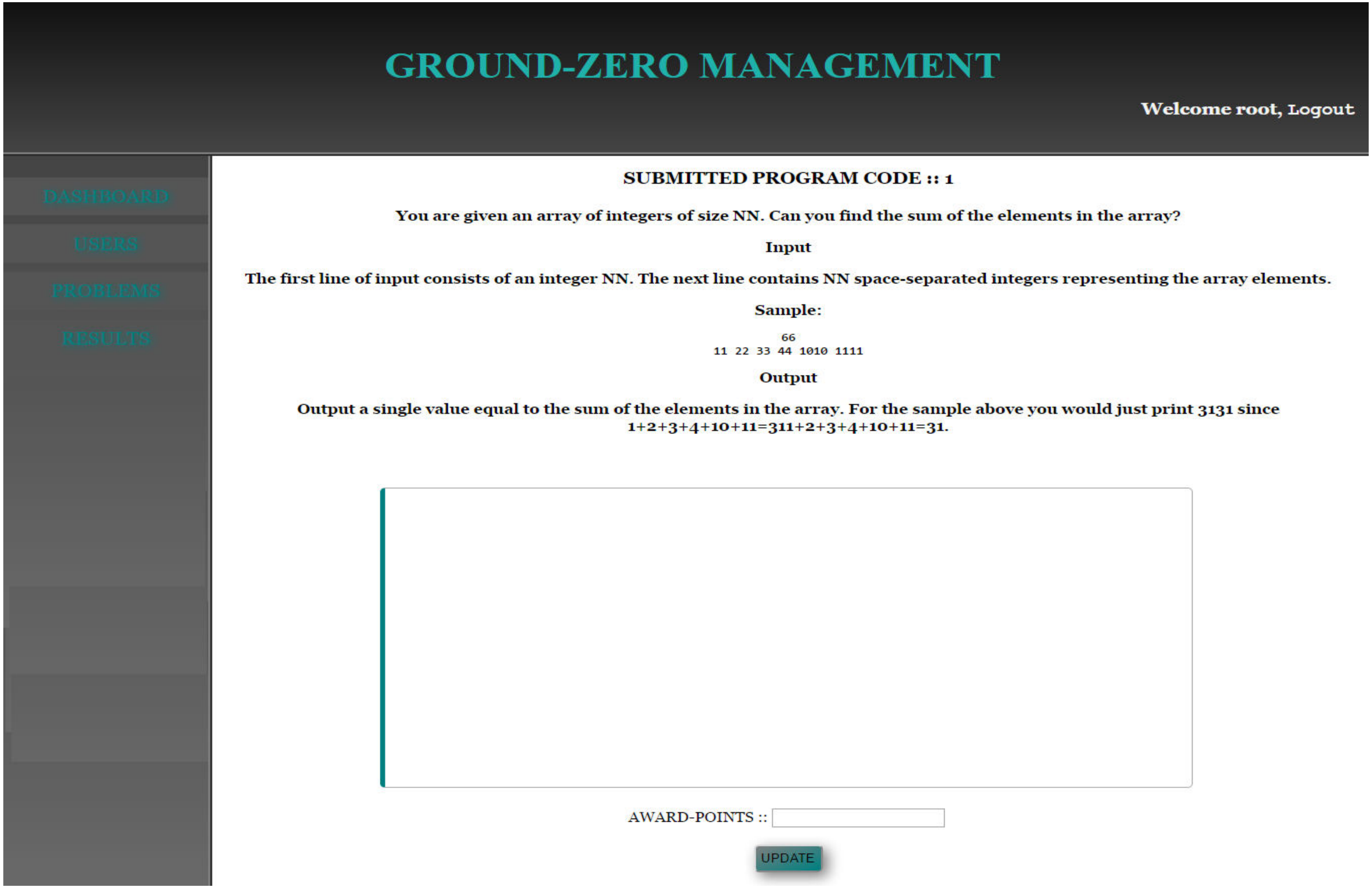


** Select A User To Delete **

USER-NAME	FULL-NAME	SELECT
adi.singh1992	Aditya	○
shiv2016	Shiv Mohan	○
shivam2016	Shivam Verma	○



EASY:: ○ INTERMEDIATE:: ○ EXPERT:: ○



SUBMITTED PROGRAM CODE :: 1

You are given an array of integers of size NN. Can you find the sum of the elements in the array?

Input

The first line of input consists of an integer NN. The next line contains NN space-separated integers representing the array elements.

Sample:

```
66
11 22 33 44 1010 1111
```

Output

Output a single value equal to the sum of the elements in the array. For the sample above you would just print 3131 since 1+2+3+4+10+11=311+2+3+4+10+11=31.

SOURCE-CODE

Header.jsp

```
<html>

<head>

  <title>GROUND-ZERO</title>

  <link type="text/css" href="static/css/ground_zero.css" rel="stylesheet"/>

  <link type="text/css" href="static/css/layout.css" rel="stylesheet"/>

    <link type="text/css" href="static/css/problems.css" rel="stylesheet"/>

  <link type="text/css" href="static/css/top_nav.css" rel="stylesheet"/>

    <script type="text/javascript" src="static/js/javas.js"></script>

</head>

<body>

  <header class="gz-container gz-center">

    <a href="index.jsp"><img src="static/images/logo.png" style="width:56.5%"/></a>

    <h2 class="gz-center">ONLINE COMPETITIVE PROGRAMMING PORTAL <BR/> <b>PRACTICE-CHALLENGE-IMPROVE</b></h2>

  </header>
```

Footer.jsp

```
<footer class="gz-container gz-teal">
  <p class="gz-large gz-center" id="copyright">
      All Rights Reserved :: Ground-Zero Team
      <br/>
      Copyright &copy 2015 ::
      <%= new java.util.Date().getYear() + 1900 %>
  </p>
</footer>
</body>
</html>
```

Nav_bar.jsp

```
    <br/><br/> <div class="top_nav">
    <ul><li><a href="index.jsp">HOME</a></li>
       <li><a href="login.jsp">LOGIN</a></li>
       <li><a href="register.jsp">REGISTRATION</a></li>
       <li><a href="categ.jsp">COMPETITION</a></li>
       <li><a href="result.jsp">RESULTS</a></li>
       <li><a href="ranking.jsp">RANKINGS</a></li>
       <li><a href="faq.jsp">FAQ</a></li>
                 <li><a href="logout.jsp">LOGOUT</a></li>
    </ul></div>
    <br/><br/>
```

Index.jsp

```
<%@include file="/header.jsp" %>

<%  String myname =  (String)session.getAttribute("session"); %>

   <div class="gz-container">

      <img src="static/images/home_graphic.png" style="width: 46%; height: 52.5%"/>

      <div class="gz-center" id="h-card">

         <h2>ENTER THE PORTAL</h2>

         <button class="gz-btn gz-teal" id="b1" onClick="user()">USER</button>

         <button class="gz-btn gz-teal" id="b2" onClick="admin()">ADMIN</button>

         <br/><br/>

                  <%

                     if(myname!=null)

                        {

                        out.print("<h3 id='b3'>Welcome "+myname+" ..!!</h3>");

                     }

                     else

                        {

                  %>

         <h3 id="b3" onClick="register()">Not Registered..!! Join Now</h3>

                  <%

                  }
```

```
                %>
            </div>
    </div>
        <div class="float" style="background-image: 
url('static/images/faq.png');" onclick="location.href='faq.jsp';"></div>
<%@include file="/footer.jsp" %>
```

Error.jsp

```
<%@include file="/header.jsp" %>
<script language="javascript">
    setTimeout(function(){ location.href="index.jsp"; }, 5000);
</script>
        <div style="text-align: center; height: 50%;">
            <h2 style="font-family: georgia; animation: blink1 1s linear 
infinite;"><br/>
            Whoops..!! Something Went Wrong,<br/>
            You Will Be Redirected To The Home Page in Five Seconds.
            </h2>
        </div>
<%@include file="/footer.jsp" %>
```

Login.jsp

```
<%@include file="/header.jsp" %>
<%
   String myname =  (String)session.getAttribute("session");
   if(myname!=null)
      {
      response.sendRedirect("categ.jsp");
   }
   else
     {
%>
   <div class="gz-container gz-padding gz-card-8" id="log_form">
   <form action="loginprocess.jsp" method="post">
      <div class="gz-group">
         <input type="text" name="uname" class="gz-input" required/>
         <label class="gz-label">User-Name</label>
      </div>
      <div class="gz-group">
         <input type="password" name="pass" class="gz-input" required/>
         <label class="gz-label">Password</label>
      </div>
         <button type="submit" class="gz-btn gz-teal">SUBMIT</button>
```

```
        <br/><p onclick="forgot()" style="color: red; font-size:
1.12em;">FORGOT YOUR PASSWORD? Reset It Now..!!</p>
  </form>
  </div>
<%
  }
%>
<%@include file="/footer.jsp" %>
```

Logout.jsp

```
<%
  session.setAttribute("session", null);
  session.setAttribute("admin", null);
  session.invalidate();
  response.sendRedirect("index.jsp");
%>
```

Login_process.jsp

```
<%@page import="bean.Login"%>

<jsp:useBean id="obj" class="bean.LoginBean"/>

<jsp:setProperty property="*" name="obj"/>

<%

boolean status=Login.validate(obj);

if(status){

   String user = request.getParameter("uname");

   session.setAttribute("session",user);

   response.sendRedirect("categ.jsp");

}

else{

   out.print("<script>alert('Wrong User-Name or Password');</script>");

%>

<%@include file="login.jsp"%>

<%

}

%>
```

Register.jsp

```
<%@include file="/header.jsp" %>

<%

   String myname =  (String)session.getAttribute("session");

   if(myname!=null)

      {

      response.sendRedirect("categ.jsp");

   }

   else

     { %>

   <div class="gz-container gz-padding gz-card-8" id="log_form">

      <form action="process.jsp" method="POST" onsubmit="return
validateform()">

         <div class="gz-group">

            <input type="text" name="fname" id="fname" class="gz-input"
required/>

            <label class="gz-label">Full-Name</label>

         </div>

         <div class="gz-group">

            <input type="email" name="uemail" id="uemail" class="gz-input"
required/>

            <label class="gz-label">E-Mail</label>

         </div>

         <div class="gz-group">
```

```
        <input type="text" name="uname" id="uname" class="gz-input" required/>
        <label class="gz-label">User-Name</label>
      </div>
      <div class="gz-group">
        <input type="password" name="upass" id="upass"  class="gz-input" required/>
        <label class="gz-label">Password</label>
      </div>
      <div class="gz-group">
        <input type="password" name="cpass" id="cpass" class="gz-input" required/>
        <label class="gz-label">Confirm Password</label>
      </div>
      <div class="gz-group">
      <label class="gz-checkbox">
        <input type="checkbox" name="tnc" id="tnc"/>
        <div class="gz-checkmark"></div> I Agree To Ground-Zero's Terms & Conditions
      </label>
      </div>
      <button type="submit" class="gz-btn gz-teal">REGISTER</button>
    </form>
  </div> <%    }    %>
<%@include file="/footer.jsp" %>
```

Process.jsp

```
<%@page import="bean.Register"%>

<%@page import="bean.sendMail"%>

<jsp:useBean id="obj" class="bean.User"/>

<jsp:setProperty property="*" name="obj"/>

<% String email = request.getParameter("uemail");

String uname = request.getParameter("uname");

String pass = request.getParameter("upass");

int status=Register.register(obj);

if(status<0){

   out.print("<script>alert('This Username Is Already Registered, Try Another
One..');</script>");

   out.print("<script>window.history.back();</script>");

}

else if(status>0){

   bean.sendMail.sendE(email, uname, pass);

   out.print("<script>alert('Registered Successfully. Click OK to go to Login
screen');</script>");

   out.print("<script>window.location.assign('login.jsp');</script>");

}

else{

   out.print("<script>alert('System Error, Try Again In A Few
Minutes');</script>");

   out.print("<script>window.history.back();</script>");

} %>
```

Categ.jsp

```
<%
   String myname = (String)session.getAttribute("session");
   if(myname!=null){%>
      <%@include file="header.jsp"%>
      <div>
         <form action="comp.jsp" method="POST">
            <div class="c_contain">
               <button value="easy" name="type" class="gz-light-grey
t_btn"><span class="c_label">EASY</span></button>
               <button value="intermediate" name="type" class="gz-light-grey
t_btn"><span class="c_label">INTERMEDIATE</span></button>
               <button value="expert" name="type" class="gz-light-grey
t_btn"><span class="c_label">EXPERT</span></button>
               <br/><br/>
               <h3 id="b3">PLEASE SELECT A TYPE OF CHALLENGE</h3>
            </div>
         </form>
      </div>
      <%@include file="footer.jsp"%>
   <%  }
   else
      response.sendRedirect("index.jsp");
%>
```

Comp.jsp

```
<%@include file="/header.jsp" %>

<%@include file="/nav_bar.jsp" %>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%! int count; %>

<%

  String myname =  (String)session.getAttribute("session");

  Integer qcount = (Integer)session.getAttribute("count");

  String type = request.getParameter("type");

  if(type!=null){

  if(myname==null){

    response.sendRedirect("login.jsp");

  }

  else{

    if(qcount==null){

      count=1;

      session.setAttribute("count",count);

    }

    else

      count++;

    Statement st=ConnectionProvider.getCon().createStatement();

    ResultSet rs;
    rs=st.executeQuery("Select "+type+" from competition where
id="+count+"");

    String type_hash = "";

    if(type.equals("easy"))

      type_hash = "48bb6e862e54f2a795ffc4e541caed4d";

    else if(type.equals("intermediate"))

      type_hash = "438fa616dea43dbb0a42b0ce2d393e7a";

    else if(type.equals("expert"))

      type_hash = "b9b83bad6bd2b4f7c40109304cf580e1";

%>

    <div id="comp">

      <h3>PROBLEM STATEMENT :: <%=count %></h3>

      <div id="comp1">

        <%

          if(rs.next()){

            if(rs.getString(1)==null){

              session.setAttribute("count", null);

              out.print("<script>alert('Great..!! You have Solved It
All.');</script>");

              out.print("<script>location.href='categ.jsp';</script>");

              }

            out.print(rs.getString(1));

          }

          else{
```

```
                    session.setAttribute("count", null);

                    out.print("<script>alert('Great..!! You have Solved It 
All.');</script>");

                    out.print("<script>location.href='categ.jsp';</script>");

                }

            %>

          </div>

          <form method="POST" action="comp_process.jsp" class="c_form">

            <textarea name="code" spellcheck="false"></textarea>

            <input type="hidden" value="<%=type_hash %>" name="cvalue"/>

            <input type="hidden" value="<%=count %>" name="id"/>

            <input type="hidden" value="<%=myname %>" name="uname"/>

            <br/><br/>

            <input type="button" value="NEXT" onclick="location.reload()" 
class="gz-teal ps1"/>

            <input type="submit" value="SUBMIT YOUR CODE" class="gz-teal 
ps1"/>

          </form>

        </div>

<%

        }

    }

%>

<%@include file="/footer.jsp" %>
```

Comp_process.jsp

```
<%@ page import="bean.comp_process"%>

<jsp:useBean id="obj" class="bean.comp_bean"/>

<jsp:setProperty property="*" name="obj"/>


<%

   int status = comp_process.c_process(obj);

   if(status>0)

      out.print("<script>alert('Code Successfully Submitted');</script>");

   else if(status<0)

      out.print("<script>alert('Awww..!! Snap..Something Went Wrong. Please
Try Again.');</script>");

   else

      out.print("<script>alert('Database Error. Please Try Again.');</script>");

   out.print("<script>location.href='categ.jsp';</script>");

%>
```

Result.jsp

```
<%@include file="/header.jsp" %>

<%@include file="/nav_bar.jsp" %>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%

   String myname =  (String)session.getAttribute("session");

   String points, q_code;

   if(myname==null)

      {

      response.sendRedirect("login.jsp");

   }

   else

     {

      Statement st=ConnectionProvider.getCon().createStatement();

      Statement st1=ConnectionProvider.getCon().createStatement();

      Statement st2=ConnectionProvider.getCon().createStatement();

      ResultSet rs,ps,ss;

      rs=st.executeQuery("Select * from easy_result where
uname='"+myname+"' order by id asc");

      ps=st1.executeQuery("Select * from intermediate_result where
uname='"+myname+"' order by id asc");

      ss=st2.executeQuery("Select * from expert_result where
uname='"+myname+"' order by id asc");

%>

      <div class="categ1">
```

```
    <ul>
        <li onclick="show_hide3()">EXPERT</li>
        <li onclick="show_hide2()">INTERMEDIATE</li>
        <li onclick="show_hide1()">EASY</li>
    </ul>
</div>
<div id="easy" class="r_contain">
    <h3>Results For Competition (EASY)</h3>
    <table class="gz-table gz-striped gz-border gz-bordered r_table">
        <tr class="gz-teal">
            <td>Question Code</td>
            <td>Points Awarded</td>
        </tr>
        <%
        while(rs.next()){
            q_code=rs.getString(1);
            points=rs.getString(4);
            %>
        <tr>
            <td><%=q_code%></td>
            <td><%=points%></td>
        </tr>
        <%
        }
    %>
    </table>
```

```
</div>
<div id="inter" class="r_contain">
  <h3>Results For Competition (INTERMEDIATE)</h3>
  <table class="gz-table gz-striped gz-border gz-bordered r_table">
    <tr class="gz-teal">
      <td>Question Code</td>
      <td>Points Awarded</td>
    </tr>
    <%
    while(ps.next()){
      q_code=ps.getString(1);
      points=ps.getString(4);
      %>
    <tr>
      <td><%=q_code%></td>
      <td><%=points%></td>
    </tr>
      <%
    }
  %>
  </table>
</div>

<div id="expert" class="r_contain">
  <h3>Results For Competition (EXPERT)</h3>
  <table class="gz-table gz-striped gz-border gz-bordered r_table">
```

```
          <tr class="gz-teal">
            <td>Question Code</td>
            <td>Points Awarded</td>
          </tr>
          <%
          while(ss.next()){
            q_code=ss.getString(1);
            points=ss.getString(4);  %>
          <tr>
            <td><%=q_code%></td>
            <td><%=points%></td>
          </tr>
            <%   }
      }    %>
        </table>
      </div>
<%@include file="/footer.jsp" %>
```

Ranking.jsp

```
<%@include file="/header.jsp" %>

<%@include file="/nav_bar.jsp" %>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%

  String myname =  (String)session.getAttribute("session");

  if(myname==null)

    {

    response.sendRedirect("login.jsp");

  }

  else

   {

    Statement st=ConnectionProvider.getCon().createStatement();

    Statement st1=ConnectionProvider.getCon().createStatement();

    Statement st2=ConnectionProvider.getCon().createStatement();

    ResultSet rs,ps,ss;

    rs=st.executeQuery("select uname, sum(points) from easy_result group by
uname;");

    ps=st1.executeQuery("select uname, sum(points) from intermediate_result
group by uname;");

    ss=st2.executeQuery("select uname, sum(points) from expert_result group
by uname;");

%>
```

```
<h1 class="r_contain1">Ground-Zero Rankings</h1>
<div class="r_contain2">
      <br/><br/>
      <table class="gz-table gz-striped gz-border gz-bordered r_table">
        <caption>EASY MODE</caption>
        <tr class="gz-teal">
          <td>User Name</td>
          <td>Total Points</td>
        </tr>
        <%
        while(rs.next()){
          %>
        <tr>
          <td><%=rs.getString(1)%></td>
          <td><%=rs.getString(2)%></td>
        </tr>
        <%
        }
      %>
      </table>
      <br/><br/>
      <table class="gz-table gz-striped gz-border gz-bordered r_table">
        <caption>INTERMEDIATE MODE</caption>
```

```
    <tr class="gz-teal">
        <td>User Name</td>
        <td>Total Points</td>
    </tr>
    <%
    while(ps.next()){
        %>
    <tr>
        <td><%=ps.getString(1)%></td>
        <td><%=ps.getString(2)%></td>
    </tr>
    <%
    }
%>
</table>
<br/><br/>
<table class="gz-table gz-striped gz-border gz-bordered r_table">
    <caption>EXPERT MODE</caption>
    <tr class="gz-teal">
        <td>User Name</td>
        <td>Total Points</td>
    </tr>
    <%
```

```
        while(ss.next()){
          %>
        <tr>
          <td><%=ss.getString(1)%></td>
          <td><%=ss.getString(2)%></td>
        </tr>
        <%
        }
      %>
      </table>
      <br/><br/>
    </div>
<%
    }
%>
<%@include file="/footer.jsp"%>
```

Faq.jsp

```
<%@include file="/header.jsp" %>

<%@include file="/nav_bar.jsp" %>

<div class="faq">

    <h3>Frequently Asked Questions :-</h3>

    <br/>

    <p class="q">Q: What is Ground-Zero?</p>

    <p class="a">A: Ground-Zero is an online competitive programming
portal.</p>

    <br/>

    <p class="q">Q: What are it's benefits?</p>

    <p class="a">A: It allows a user to practice programming problems at his
convenience.</p>

    <br/>

    <p class="q">Q: What is the level of the problems?</p>

    <p class="a">A: The problems are divided into a sub-category of Easy,
Intermediate and Expert level.</p>

    <br/>

    <p class="q">Q: How will i get scored?</p>

    <p class="a">A: For now, the codes submitted by users will be checked by
an administrator who will be awarding points.</p>

    <br/>

    <p class="q">Q: How many points can i get?</p>
    <p class="a">A: The points will be awarded on a scale of 10, 10 being the maximum and 1 being the lowest.</p>

    <br/>

    <p class="q">Q: Can i submit multiple codes for a single question?</p>

    <p class="a">A: No, you can submit only one solution code for a problem. The points will be awarded on the first code submitted.</p>

    <br/>

    <p class="q">Q: Why cannot i submit multiple solution codes for the same problem?</p>

    <p class="a">A: This version of the application currently does not support this functionality, it will be added in future.</p>

</div>

<%@include file="/footer.jsp" %>
```

Admin_login.jsp

```
<%@include file="/header.jsp" %>

<%

   String myname =  (String)session.getAttribute("admin");

   if(myname!=null)

      {

      response.sendRedirect("mgmnt.jsp");

   }

   else  {  %>

   <div class="gz-container gz-padding gz-card-8" id="log_form"
style="height:41.5%">

   <form action="admin_process.jsp" method="post">

      <div class="gz-group">

         <input type="text" name="aname" class="gz-input" required/>

         <label class="gz-label">Admin-Name</label></div>

      <div class="gz-group">

         <input type="password" name="pass" class="gz-input" required/>

         <label class="gz-label">Password</label>

      </div>

      <button type="submit" class="gz-btn gz-teal">SUBMIT</button>

   </form></div>

<%  } %>

<%@include file="/footer.jsp" %>
```

Admin_process.jsp

```
<%@page import="bean.aLogin"%>

<jsp:useBean id="obj" class="bean.aLoginBean"/>

<jsp:setProperty property="*" name="obj"/>

<%

boolean status=aLogin.validate(obj);

if(status){

   String user = request.getParameter("aname");

   session.setAttribute("admin",user);

   response.sendRedirect("mgmnt.jsp");

}

else{

   out.print("<script>alert('Wrong Admin-Name or Password');</script>");

%>

<%@include file="admin_login.jsp"%>

<%

}

%>
```

Forgot.jsp

```
<%@ page import="java.sql.*" %>

<%@ page import="bean.Hash" %>

<%@ page import="bean.ConnectionProvider" %>

<%

   String uname = request.getParameter("uname");

   String email = request.getParameter("email");

   String pass = request.getParameter("pass");

   if(uname.trim().equals("") || email.trim().equals("") ||
pass.trim().equals("")){

      out.print("Empty Fields");

   }

   else{

      pass = Hash.generateMD5(pass);

      String uemail = "";

      Connection con = ConnectionProvider.getCon();

      Connection con1 = ConnectionProvider.getCon();

      PreparedStatement st = con.prepareStatement("select email from
userbase where uname=?;");

      st.setString(1,uname);

      ResultSet rs = st.executeQuery();

      if(rs.next()){

         uemail = rs.getString(1);
```

```
            PreparedStatement st1 = con1.prepareStatement("update userbase set
pass=? where uname=?");
            st1.setString(1,pass);
            st1.setString(2,uname);
            int x = st1.executeUpdate();
            if(x == 0)
                out.print("Password Reset Failed..!!");
            else
                out.print("Password Updation Successfull..!!");
        }
        else
            out.print("No Such User Exists..!!");
    }
%>
```

Dash_head.jsp

```
<%
   String myname =  (String)session.getAttribute("admin");
   if(myname==null)
      {
      response.sendRedirect("admin_login.jsp");
   }
   else
     {
%>
<html>
   <head>
      <title>Ground-Zero | C.M.S.</title>
      <link href="static/css/admin_style.css" type="text/css" rel="stylesheet"/>
   </head>
   <body>
      <div class="head">
         <h1 class="title">GROUND-ZERO MANAGEMENT</h1>
         <h3 class="aname">Welcome <%=myname%>, <a
href="logout.jsp">Logout</a></h3>
      </div>
      <div class="side_nav">
         <ul>
```

```
          <li><a href="mgmnt.jsp">DASHBOARD</a></li>
          <li><a href="users.jsp">USERS</a></li>
          <li><a href="mod_probs.jsp">PROBLEMS</a></li>
          <li><a href="update_res.jsp">RESULTS</a></li>
        </ul>
      </div>
```

Mgmnt.jsp

```
<%@include file="dash_head.jsp"%>
      <div class="mgmnt">
        <a href="users.jsp"><img src="static/images/users.png" alt="manage users"/></a>
        <a href="mod_probs.jsp"><img src="static/images/problems.png" alt="update problems"/></a>
        <a href="update_res.jsp"><img src="static/images/results.png" alt="update results"/></a>
      </div>
    </body>
</html>
<%
      }
%>
```

Users.jsp

```
<%@include file="/dash_head.jsp" %>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<script>

  function confirm1(){

    var x = confirm("Are You Sure You Want To Delete This User?");

    if(x==true)

      return true;

    else

      return false;

  }

</script>

  <div class="contain">

  <div class="users">

    <h2>** Select A User To Delete **</h2>

    <form action="user_process.jsp" method="post" onsubmit="return
confirm1();">

      <table>

        <tr>

          <th>USER-NAME</th>

          <th>FULL-NAME</th>

          <th>SELECT</th>
```

```
            </tr>
      <%
         Statement st=ConnectionProvider.getCon().createStatement();
         ResultSet rs;
         rs=st.executeQuery("Select uname,fname from userbase;");
         while(rs.next()){
      %>
            <tr>
               <td><%=rs.getString(1)%></td>
               <td><%=rs.getString(2)%></td>
               <td><input type="radio" name="users"
   value="<%=rs.getString(1)%>"/></td>
            </tr>
         <%   }   %>
          </table>
            <br/>
            <input type="submit" value="DELETE" class="us"/>
        </form>
      </div>
      </div>
      </body>
   </html>
   <%   } %>
```

User_process.jsp

```
<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%

  String user = request.getParameter("users");

  Statement st=ConnectionProvider.getCon().createStatement();

  try{

    st.executeUpdate("delete from userbase where uname='"+user+"';");

    out.print("<script>alert('User Successfully Deleted');</script>");

    out.print("<script>location.href='mgmnt.jsp';</script>");

  }catch(Exception e){

      out.print(e.getMessage());

      }

%>
```

Mod_probs.jsp

```
<%@include file="dash_head.jsp"%>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<form action="probs_process.jsp" method="post" class="c_form">

        <br/><br/>

        EASY:: <input type="radio" value="easy" name="type"/>

        INTERMEDIATE:: <input type="radio" value="intermediate"
name="type"/>

        EXPERT:: <input type="radio" value="expert" name="type"/>

        <br/><br/>

        <textarea name="code" spellcheck="false"></textarea>

        <br/><br/>

        <input type="submit" value="UPDATE" class="us"/>

      </form>

   </body>

</html>

<%

     }

%>
```

Probs_process.jsp

```
<%@ page import="bean.probs_process"%>
<jsp:useBean id="obj" class="bean.probs_bean"/>
<jsp:setProperty property="*" name="obj"/>

<%
  int status = probs_process.p_process(obj);
  if(status>0)
    out.print("<script>alert('Problem Added Successfully');</script>");
  else if(status<0)
    out.print("<script>alert('Awww..!! Snap..Something Went Wrong. Please Try Again.');</script>");
  else
    out.print("<script>alert('Database Error. Please Try Again.');</script>");
  out.print("<script>location.href='mod_probs.jsp';</script>");
%>
```

Update_res.jsp

```
<%@include file="/dash_head.jsp" %>
<%@ page import="java.sql.*" %>
<%@ page import="bean.ConnectionProvider" %>
  <div class="contain">
  <div class="users">
    <h2>** Select A User To Update Result **</h2>
    <form action="res_process.jsp" method="post">
      <table>
        <tr>
          <th>USER-NAME</th>
          <th>FULL-NAME</th>
          <th>SELECT</th>
        </tr>
  <%
    Statement st=ConnectionProvider.getCon().createStatement();
    ResultSet rs;
    rs=st.executeQuery("Select uname,fname from userbase;");
    while(rs.next()){
  %>
        <tr>
          <td><%=rs.getString(1)%></td>
          <td><%=rs.getString(2)%></td>
          <td><input type="radio" name="users"
value="<%=rs.getString(1)%>"/></td>
        </tr>
    <%
      }
    %>
      </table>
        <br/>
        EASY :: <input type="radio" name="type" value="easy"/>
        INTERMEDIATE :: <input type="radio" name="type"
value="intermediate"/>
        EXPERT :: <input type="radio" name="type" value="expert"/>
        <br/><br/>
        <input type="submit" value="SUBMIT" class="us"/>
    </form>
  </div>
  </div>
  </body>
</html>
<%
  }
%>
```

Res_process.jsp

```
<%@include file="/dash_head.jsp" %>

<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%!int count = 0;%>

<%

   String user = request.getParameter("users");

   String type = request.getParameter("type");

   String query = "", q = "";

   Integer qcount = (Integer)session.getAttribute("count");

   if(qcount==null){

      count=1;

      session.setAttribute("count",count);

   }

   else

      count++;

   if(type.equals("easy")){

      query = "Select code FROM EASY_RESULT WHERE id="+count+" AND
uname='"+user+"';";

      q = "Select easy from competition where id="+count+";";

   }

   else if(type.equals("intermediate")){
    query = "Select code FROM INTERMEDIATE_RESULT WHERE id="+count+" AND uname='"+user+"';";
    q = "Select intermediate from competition where id="+count+";";
  }
  else if(type.equals("expert")){
    query = "Select code FROM EXPERT_RESULT WHERE id="+count+" AND uname='"+user+"';";
    q = "Select easy from competition where id="+count+";";
  }
  Statement s=ConnectionProvider.getCon().createStatement();
  Statement st=ConnectionProvider.getCon().createStatement();
  try{
    ResultSet r = s.executeQuery(q);
    ResultSet rs = st.executeQuery(query);
%>
    <div class="sub_probs">
      <h3>SUBMITTED PROGRAM CODE :: <%=count %></h3>
        <%
          if(rs.next() && r.next()){
        %>
      <h4><%=r.getString(1) %></h4>
      <form method="POST" action="res_ins.jsp">
        <br/><br/>
        <textarea name="code"
spellcheck="false"><%=rs.getString(1)%></textarea>

        <br/><br/>

        <input type="hidden" value="<%=count %>" name="id"/>

        <input type="hidden" value="<%=user %>" name="uname"/>

        <input type="hidden" value="<%=type %>" name="type"/>

        AWARD-POINTS :: <input type="number" name="points"/>

        <br/><br/>

        <input type="submit" value="UPDATE" class="us"/>

      </form>

    </div>

<%    }

          else{

            session.setAttribute("count", null);

            out.print("<script>alert('Ooops..!! No More Problems For This
User.');</script>");

            out.print("<script>location.href='update_res.jsp';</script>");

          }

  }catch(Exception e){

      out.print(e.getMessage());

      }    %>

  </body>

</html>

<%    }    %>
```

Res_ins.jsp

```
<%@ page import="java.sql.*" %>

<%@ page import="bean.ConnectionProvider" %>

<%  Integer id = Integer.parseInt(request.getParameter("id"));

   String user = request.getParameter("uname");

   String type = request.getParameter("type");

   Integer points = Integer.parseInt(request.getParameter("points"));

   String query = "";

   if(type.equals("easy"))

      query = "UPDATE EASY_RESULT SET points="+points+" WHERE id="+id+"
and uname='"+user+"';";

   else if(type.equals("intermediate"))

         query = "UPDATE INTERMEDIATE_RESULT SET points="+points+" WHERE
id="+id+" and uname='"+user+"';";

   else if(type.equals("expert"))

         query = "UPDATE EXPERT_RESULT SET points="+points+" WHERE
id="+id+" and uname='"+user+"';";

PreparedStatement ps =
ConnectionProvider.getCon().prepareStatement(query);

   int x = 0;

   x  = ps.executeUpdate();

   if(x==0){

      out.print("<script>alert('Error..!! Please Try Again');</script>");

      out.print("<script>window.history.back();</script>");

   }   else{

      out.print("<script>alert('Points Awarded Successfully');</script>");

      out.print("<script>location.href='update_res.jsp';</script>");  }  %>
```

ConnectionProvider.java

```
package bean;

import java.sql.*;

import static bean.Provider.*;


public class ConnectionProvider {

private static Connection con=null;

static{

try{

Class.forName(DRIVER);

con=DriverManager.getConnection(CONNECTION_URL,USERNAME,PASSWORD
);

}catch(Exception e){}

}


public static Connection getCon(){

   return con;

}

}
```

Provider.java

```
package bean;

public interface Provider {

String DRIVER="com.mysql.jdbc.Driver";

String CONNECTION_URL="jdbc:mysql://localhost:3306/project";

String USERNAME="root";

String PASSWORD="password";

}
```

Register.java

```
package bean;

import java.sql.*;


public class Register{

public static int register(User u){

int status=0;

try{

Connection con=ConnectionProvider.getCon();

PreparedStatement ch=con.prepareStatement("select uname from userbase
where uname=?");

ch.setString(1,u.getUname());

ResultSet rs=ch.executeQuery();
if(rs.next()){

   status = -1;

}

else{

PreparedStatement ps=con.prepareStatement("insert into userbase
values(?,?,?,?)");

String pwd=Hash.generateMD5(u.getUpass());

ps.setString(1,u.getFname());

ps.setString(2,u.getUemail());

ps.setString(3,u.getUname());

ps.setString(4,pwd);

status=ps.executeUpdate();

}

}catch(Exception e){}

return status;

}

}
```

User.java

```
package bean;
public class User {
private String fname,uname,upass,uemail;
public String getFname() {
   return fname;
}
public void setFname(String fname) {
   this.fname = fname;
}
public String getUname() {
   return uname;
}
public void setUname(String uname) {
   this.uname = uname;
}
public String getUpass() {
   return upass;
}
public void setUpass(String upass) {
   this.upass = upass;
}
public String getUemail() {
   return uemail;
}
public void setUemail(String uemail) {
   this.uemail = uemail;
} }
```

Login.java

```
package bean;
import java.sql.*;
public class Login{
   public static boolean status;
   public static boolean validate(LoginBean bean){
      try{
         Connection con=ConnectionProvider.getCon();
         PreparedStatement ps=con.prepareStatement("select * from userbase
where uname=? and pass=?");
         String pwd=Hash.generateMD5(bean.getPass());
                  ps.setString(1,bean.getUname());
         ps.setString(2, pwd);
         ResultSet rs=ps.executeQuery();
                  status = rs.next();
            }catch(Exception e){}
      return status;
   }
}
```

LoginBean.java

```
package bean;

public class LoginBean {

private String uname,pass;

public String getUname() {

   return uname;

}

public void setUname(String uname) {

   this.uname = uname;

}

public String getPass() {

   return pass;

}

public void setPass(String pass) {

   this.pass = pass;

}

}
```

ALogin.java

```
package bean;

import java.sql.*;

public class aLogin{

    public static boolean status;

    public static boolean validate(aLoginBean bean){

        try{

            Connection con=ConnectionProvider.getCon();

            PreparedStatement ps=con.prepareStatement("select * from adminbase where aname=? and pass=?");

            String pwd=Hash.generateMD5(bean.getPass());

                        ps.setString(1,bean.getAname());

            ps.setString(2, pwd);

            ResultSet rs=ps.executeQuery();

                        status = rs.next();

                }catch(Exception e){}

        return status;

    }

}
```

AloginBean.java

```
package bean;


public class aLoginBean {
private String aname,pass;


public String getAname() {
   return aname;
}
public void setAname(String aname) {
   this.aname = aname;
}
public String getPass() {
   return pass;
}
public void setPass(String pass) {
   this.pass = pass;
}
}
```

Hash.java

```
package bean;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class Hash {
   private Hash() {
}
public static String generateMD5(String message){
   return hashString(message, "MD5");
}
private static String hashString(String message, String algorithm){
      try {
         MessageDigest digest = MessageDigest.getInstance(algorithm);
         byte[] hashedBytes = digest.digest(message.getBytes("UTF-8"));
         return convertByteArrayToHexString(hashedBytes);
      }
      catch (NoSuchAlgorithmException | UnsupportedEncodingException ex){}
      return message;
}
private static String convertByteArrayToHexString(byte[] arrayBytes) {
   StringBuffer stringBuffer = new StringBuffer();
   for (int i = 0; i < arrayBytes.length; i++) {
      stringBuffer.append(Integer.toString((arrayBytes[i] & 0xff) + 0x100,
16).substring(1));
   } return stringBuffer.toString();   }   }
```

SendMail.java

```
package bean;

import java.util.Properties;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.PasswordAuthentication;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;


public class sendMail{

  public static void sendE(String email, String uname, String pass) {

    Properties props = new Properties();

    props.put("mail.smtp.host", "smtp.gmail.com");

    props.put("mail.smtp.socketFactory.port", "465");

    props.put("mail.smtp.socketFactory.class",
"javax.net.ssl.SSLSocketFactory");

    props.put("mail.smtp.auth", "true");

    props.put("mail.smtp.port", "465");

    Session session;

        session = Session.getDefaultInstance(props, new
javax.mail.Authenticator(){
```

```
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
              return new
PasswordAuthentication("adi.singh1992","2don2dtitq");
            }
          });
    try {
      Message message = new MimeMessage(session);
      message.setFrom(new InternetAddress("adi.singh1992@gmail.com"));
          message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(email));
          message.setSubject("Welcome To Ground-Zero");
          message.setText("Hey There, \n\nWelcome to Ground-Zero. "
              + "Below are your Login details for future reference.\n\n User-
Name: "+uname+" \n\n "
              + "Password: "+pass+"");
          Transport.send(message);
    }catch(MessagingException e){
      throw new RuntimeException(e);
    }
  }
}
```

Comp_process.java

```
package bean;
import java.sql.*;
public class comp_process{
        public static int c_process(comp_bean b){
                int status = 0;
                try{
                    Connection con = ConnectionProvider.getCon();
                    PreparedStatement ps = con.prepareStatement(getQuery(b));
                    ps.setString(1,b.getId());
                    ps.setString(2,b.getUname());
                    ps.setString(3,b.getCode());
                    status = ps.executeUpdate();
                }catch(Exception e){}
                return status;
        }
        public static String getQuery(comp_bean b){
            String q = " ";
            switch (b.getCvalue()) {
                case "48bb6e862e54f2a795ffc4e541caed4d":
                    q = "INSERT INTO easy_result(id,uname,code) VALUE(?,?,?)";
                    break;
                case "438fa616dea43dbb0a42b0ce2d393e7a":
```

```
          q = "INSERT INTO intermediate_result(id,uname,code)
VALUE(?,?,?)";
          break;
        case "b9b83bad6bd2b4f7c40109304cf580e1":
          q = "INSERT INTO expert_result(id,uname,code) VALUE(?,?,?)";
          break;
      }
      return q;
    }
}
```

Comp_Bean.java

```
package bean;
public class comp_bean{
  private String code, c_value, uname, id;


  public void setCode(String code){
      this.code = code;
  }
  public String getCode(){
      return code;
  }
```

```
    public void setCvalue(String cvalue){
            this.c_value = cvalue;
    }
    public String getCvalue(){
        return c_value;
    }

    public void setUname(String uname){
        this.uname = uname;
    }
    public String getUname(){
        return uname;
    }

    public void setId(String id){
        this.id = id;
    }
    public String getId(){
        return id;
    }
}
```

Probs_process.java

```
package bean;

import java.sql.*;

public class probs_process{

    public static int p_process(probs_bean b){

        int status = 0, count=0, count1=0;

        Connection con = ConnectionProvider.getCon();

        try{

          String type = b.getType();

          String code = b.getCode();

          //Getting Last ID

          Statement st1=con.createStatement();

          String query1 = "SELECT ID FROM competition";

          ResultSet rs1 = st1.executeQuery(query1);

          while(rs1.next()){

            count1 = rs1.getInt(1);

          }

          count1++;

          PreparedStatement ss = con.prepareStatement("INSERT INTO
competition(ID) value("+count1+");");

          ss.executeUpdate();

          //Getting Last ID(Column-Wise)

          Statement st=con.createStatement();
```

```
                    String query = "SELECT ID FROM competition GROUP BY "+type+"
ORDER BY ID DESC LIMIT 1";

                    ResultSet rs = st.executeQuery(query);

                    while(rs.next()){

                        count = rs.getInt(1);

                    }

                    //Inserting Data

                    String q = "UPDATE competition SET "+type+"='"+code+"' WHERE
id="+count+";";

                    PreparedStatement ps = con.prepareStatement(q);

                    status = ps.executeUpdate();

                }catch(Exception e){}

                return status;

        }

}
```

Probs_bean.java

```
package bean;


public class probs_bean{
   private String code, type;


   public void setCode(String code){
         this.code = code;
   }
   public String getCode(){
         return code;
   }


   public void setType(String type){
      this.type = type;
   }
   public String getType(){
      return type;
   }
}
```

Layout.css

```
@font-face{
   font-family: chiller;
   src: url(chiller.ttf) format("truetype");
}
.float{
  height: 151px;
  width: 59px;
  position: fixed;
  bottom: 0;
  right: 50;
  cursor: pointer;
  opacity: .8;
  transition: .3s linear
}
.float:hover{
   box-shadow: 1px 1px 10px #fff;
}
h2{
   font-family: courier;
}
#log_form input{
   width: 80%;
}
#log_form button{
   border-radius: 5px;
}
#log_form{
   margin-left: 5%;
   margin-right: 50%;
   margin-top: 2.5%;
   margin-bottom: 2.5%;
   border-radius: 10px;
}
a{
   text-decoration: none;
   color: #DF0101;
   text-shadow: 4px 2px 6px #000;
   transition: .5s;
}
a:hover{
   color: #088A68;
   text-decoration: underline;
}
p{
   font-family: courier;
}
#h-card{
   float: right;
   margin-right: 10%;
   margin-top: 2%;
```

```
}
@keyframes blinker {
   0% { opacity: 1.0; }
   50% { opacity: 0.0; }
   100% { opacity: 1.0; }
}
@keyframes blink1 {
   0% { color: teal; }
   50% { color: #ff0000; }
   100% { color: teal; }
}
#h-card h2{
   text-align: center;
   font-family: chiller;
   font-size: 60;
   animation: blinker 4s linear infinite;
}
#h-card h3{
   text-align: center;
   cursor: pointer;
   animation: blink1 1s linear infinite;
}
#h-card button{
   height: 8%;
   width: 25%;
   border-radius: 10px;
   position: relative;
}
#b1{
   animation: anime 2s linear infinite alternate;
}
#b2{
   animation: anime1 2s linear infinite alternate;
}
@keyframes anime{
   0% {left:-60px;}
   100%{left:0px;transform:rotate(90deg);}
}
@keyframes anime1{
   0% {left:60px;}
   100%{left:0px;transform:rotate(-90deg);}
}
.t_btn{
   width: 200px;
   height: 200px;
   margin: 5% 5% 2% 5%;
   border: 2px;
   border-radius: 50%;
   box-shadow: 0px 20px 40px grey;
   background: radial-gradient(#ffffff, #E8E8E8, #B8B8B8);
}
.t_btn:hover{
  animation: btn_anime .5s ease-in-out 1;
}
@keyframes btn_anime{
  0% {transform: rotate(0deg);}
  100% {transform: rotate(360deg);}
}
.c_label{
  font-family: chiller;
  font-size: 25px;
}
.t_btn:hover .c_label{
  font-size: 35px;
}
.c_contain{
  text-align: center;
}
.c_contain h3{
  font-family: courier new;
  font-weight: bold;
  animation: blinker 2s linear infinite alternate;
}
.r_contain{
  margin: 5%;
  text-align: center;
}
.r_contain1{
  font-family: courier new;
  font-weight: bold;
  text-align: center;
  margin-top: 2%;
  color: teal;
}
.r_contain2{
  font-family: georgia;
  font-size: 25px;
}
.r_table{
  width: 60%;
  text-align: center;
  margin: auto;
}
.categ1{
  margin: 1%;
  font-family: georgia;
}
.categ1 li{
  box-shadow: 0px 5px 10px grey;
  background: #e8e8e8;
  border: 1px solid #bbb;
  display: inline-block;
  padding: 1%;
  opacity: .85;
```

```
    float: right;
    cursor: pointer;
}
.categ1 li:hover{
    opacity: 1;
}
#inter{
    display: none;
}
#expert{
    display: none;
}
.faq{
    margin: 2%;
}
.faq h3{
    font-family: Copperplate Gothic Light, sans-serif;
}
.q{
    font-weight: bold;
    font-size: 22px;
    font-family: Copperplate Gothic Light, sans-serif;
    padding-left: 2%;
}
.a{
    font-weight: bold;
    font-size: 20px;
    font-family: Copperplate Gothic Light, sans-serif;
    padding-left: 2%;
    color: teal;
}
```

Top_nav.css

```
.top_nav ul{
    margin: 0 auto;
    padding: 0;
    display: table;
    overflow: hidden;
    list-style-type: none;
}
.top_nav li{
    float: left;
    margin: 0px 10px 0px 10px;
    display: block;
    border-style: solid;
    border-width: 2px 0px 2px 0px;
    border-color: teal;
    border-radius: 5px 5px 0px 0px;
}
.top_nav li:first-child{
    border-style: solid;
    border-width: 2px 0px 2px 2px;
    border-color: teal;
    border-radius: 35px 0px 0px 35px;
}
.top_nav li:last-child{
    border-style: solid;
    border-width: 2px 2px 2px 0px;
    border-color: teal;
    border-radius: 0px 35px 35px 0px;
}
.top_nav li a{
    color: black;
    display: block;
    padding: 14px 16px;
    text-align: center;
    font-family: georgia;
    text-decoration: none;
       text-shadow: none;  }
.top_nav li a:hover{
    text-shadow: 2px 2px 20px teal;  }
.top_nav li a:active{
    color: red;  }
```

Problems.css

```
#comp{
	padding: 2%;
	font-family: serif;
	font-size: 1.3em;
	color: #424242;
}
#comp h3{
	font-family: courier new;
	padding: 1%;
	font-weight: bold;
	border: 1px solid #A4A4A4;
	border-radius: 5px;
	background-color: #F2F2F2;
	box-shadow: -2px 2px 20px grey;
	margin-bottom: 2.5%;
	width: 72%;
}
#comp textarea{
	margin-left: 1%;
	border-radius: 5px;
	width: 70%;
	height: 50%;
	resize: none;
	border-left: 5px solid teal;
```

```
        overflow: hidden;
}
#comp1{
        width: 50%;
        font-size: 1em;
        margin: 1%;
}
.ps1{
        border-radius: 5px;
        padding: .5%;
        box-shadow: 5px 2px 20px grey;
        background: linear-gradient(to bottom right, grey, teal);
}
.ps1:hover{
        box-shadow: 5px 5px 20px grey;
}
```

Admin_style.css

```
body{
    margin: 0;
    font-family: georgia;
}
.head{
    background: linear-gradient(to bottom, #111, #444);
    height: 25%;
    width: 100%;
    float: right;
    position: fixed;
    border-bottom: 4px ridge grey;
}
.title{
    color: lightseagreen;
    padding-top: 1.5%;
    text-align: center;
    font-size: 250%;
    font-family: papyrus;
    margin-left: 0;
    margin-bottom: 0;
    padding-left: 1%;
}
.aname{
    float: right;
    padding: 1%;
    margin-top: 0;
    color: whitesmoke;
}
.head a{
    color: whitesmoke;
    text-decoration: none;
    font-family: courier;
}
```

```
.side_nav ul{
   top: 25.6%;
   float: left;
   padding: 1% 0 0 0;
   position: fixed;
   width: 15%;
   height: 100%;
   overflow: auto;
   margin-top: 0;
   border-right: 4px ridge grey;
   background: linear-gradient(to bottom, #444, #777);
   list-style-type: none;
}
.side_nav li{
   margin: 5% 0 5% 0;
   padding: 5% 0 5% 0;
   background-color: #555;
   cursor: pointer;
}
.side_nav li a{
   text-align: center;
   text-decoration: none;
   display: block;
   font-size: 20px;
   color: teal;
   text-shadow: 2px 2px 10px;
```

```
}
.side_nav li a:hover{
   transition: all 0.3s;
   color: red;
}
.side_nav li:hover{
   background-color: #b8b8b8;
   transition: all 0.3s;
}
.side_nav li:active{
   background-color: #99ccff;
}
.mgmnt{
   width: 50%;
   position: relative;
   top: 45%;
   left: 30%;
   border: 0;
   border-radius: 10px;
   box-shadow: 10px 10px 40px #b8b8b8;
}
.mgmnt img{
   margin: 5%;
}
.mgmnt img:hover{
   transition: all 0.5s;
```

```
    border-radius: 5px;
    box-shadow: 5px 5px 20px #b8b8b8;
}
.contain{
    position: fixed;
    left: 15%;
    top: 25%;
    text-align: center;
    width: 85%;
}
.users table {
    border-collapse: collapse;
    margin: auto;
    width: 50%;
    border: 3px ridge #888;
}
.users th, td {
    text-align: left;
    padding: 5%;
}
.users tr:nth-child(even){
    background-color: #f2f2f2;
    border: 3px ridge #888;
}
.users th {
    background-color: teal;
```

```
   color: white;
}
.us{
      border-radius: 5px;
      padding: .5%;
      box-shadow: 5px 5px 20px grey;
      background: linear-gradient(to bottom right, grey, teal);
}
.us:hover{
      box-shadow: 5px 5px 40px grey;
}
.c_form{
   position: fixed;
   height: 100%;
   left: 15%;
   top: 25%;
   text-align: center;
   width: 85%;
}
.c_form textarea{
      border-radius: 5px;
      width: 70%;
      height: 50%;
      resize: none;
      border-left: 5px solid teal;
      overflow: hidden;
```

```
}
.sub_probs{
   height: 100%;
   position: absolute;
   left: 15%;
   top: 25%;
   text-align: center;
   width: 85%;
   z-index: -1;
}
.sub_probs textarea{
      border-radius: 5px;
      width: 70%;
      height: 50%;
      resize: none;
      border-left: 5px solid teal;
      overflow: hidden;
}
```

Javas.js

```
function user(){
   window.location.href="login.jsp"; }
function admin(){
   window.location.href="admin_login.jsp"; }
function register(){
   window.location.href="register.jsp"; }
function validateform(){
   var upass=document.getElementById("upass").value;
   var cpass=document.getElementById("cpass").value;
   var tnc=document.getElementById("tnc");
   if(upass !== cpass){
      alert("Passwords Do Not Match");
      return false;   }
   if(!tnc.checked){
      alert("Please Accept Terms And Conditions Before Proceeding");
      return false;   } }
function show_hide1(){
   document.getElementById("easy").style.display = "block";
   document.getElementById("inter").style.display = "none";
   document.getElementById("expert").style.display = "none";
}
function show_hide2(){
   document.getElementById("easy").style.display = "none";
   document.getElementById("inter").style.display = "block";
   document.getElementById("expert").style.display = "none";  }
function show_hide3(){
   document.getElementById("easy").style.display = "none";
   document.getElementById("inter").style.display = "none";
   document.getElementById("expert").style.display = "block";  }
function forgot(){
   var user = prompt("Enter Your UserName");
   var email = prompt("Enter Your Email To Verify");
      var pass = prompt("Enter New Password");
      var cpass = prompt("Confirm New Password");
      if(pass === cpass){
            forgot_process(user, email, pass);       }
      else {  alert("Passwords Do Not Match..!!");   }  }
function forgot_process(user, email, pass){
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function(){
            if(xhttp.readyState === 4 && xhttp.status === 200){
                  alert(xhttp.responseText);
            }  };
      xhttp.open("GET",
"forgot.jsp?uname="+user+"&email="+email+"&pass="+pass, true);
      xhttp.send();
}
```

Web.xml

```
<web-app>
   <error-page>
      <error-code>404</error-code>
      <location>/error.jsp</location>
   </error-page>
   <error-page>
      <error-code>500</error-code>
      <location>/error.jsp</location>
   </error-page>
   <error-page>
      <exception-type>java.lang.Throwable</exception-type>
      <location>/error.jsp</location>
   </error-page>
   <error-page>
      <exception-type>java.lang.Exception</exception-type>
      <location>/error.jsp</location>
   </error-page>
</web-app>
```

Database_Creator.sql

```
CREATE DATABASE  IF NOT EXISTS `ground_zero` /*!40100 DEFAULT
CHARACTER SET utf8 */;

USE `ground_zero`;

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;

/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

---- Table structure for table `adminbase`--

DROP TABLE IF EXISTS `adminbase`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `adminbase` (

  `aname` varchar(50) DEFAULT NULL,
  `pass` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
---- Table structure for table `competition`--
DROP TABLE IF EXISTS `competition`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `competition` (
  `id` int(11) DEFAULT NULL,
  `easy` longtext,
  `intermediate` longtext,
  `expert` longtext
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
---- Table structure for table `easy_result`--
DROP TABLE IF EXISTS `easy_result`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `easy_result` (
  `id` int(11) DEFAULT NULL,
  `uname` varchar(50) DEFAULT NULL,
  `code` text,
  `points` int(11) DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*!40101 SET character_set_client = @saved_cs_client */;

---- Table structure for table `expert_result`--

DROP TABLE IF EXISTS `expert_result`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `expert_result` (

  `id` int(11) DEFAULT NULL,

  `uname` varchar(50) DEFAULT NULL,

  `code` text,

  `points` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*!40101 SET character_set_client = @saved_cs_client */;

---- Table structure for table `intermediate_result`--

DROP TABLE IF EXISTS `intermediate_result`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `intermediate_result` (

  `id` int(11) DEFAULT NULL,

  `uname` varchar(50) DEFAULT NULL,

  `code` text,

  `points` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;

---- Table structure for table `userbase`--

DROP TABLE IF EXISTS `userbase`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `userbase` (

  `FNAME` varchar(100) DEFAULT NULL,

  `EMAIL` varchar(100) DEFAULT NULL,

  `UNAME` varchar(50) NOT NULL,

  `PASS` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`UNAME`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*!40101 SET character_set_client = @saved_cs_client */;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

TESTING

Test Case Id	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	**Check User login with valid data**	**1. Go to the site.** **2. Enter user id.** **3. Enter password** **4. Click Submit**	**User id = 'user'** **Password = 'pass'**	**User should be able to successfully login into the application.**	**As Expected**	**Pass**
T2	**Check User login with invalid data**	**1. Call test case T1 with invalid data.**	**User id = 'abcd'** **Password = 'xyz'**	**Application should throw an error 'Wrong Password'.**	**As Expected**	**Pass**
T3	**Check invalid page redirection**	**1. Go to the site.** **2. Enter invalid URL after the domain name.**	**URL = 'www.ground-zero.com/abcd'**	**Application should redirect to a**	**As Expected**	**Pass**
T4	**Register user with invalid data**	**1. Go to the site.** **2. Enter invalid user data.** **3. Click Submit.**	**Email = "abcd.xyz"** **Pass = "abcd"** **Confirm Pass = "xyz"**	**Application should throw an error about password not matching and invalid email.**	**As Expected**	**Pass**
T5	**Check Password Reset Utility**	**1. Enter user-name & email.** **2. Type new pass.**	**Username = "xyz"** **Email = abc@xyz.com** **Pass = "temp"**	**The password should be successfully changed**	**As Expected**	**Pass**

SECURITY IMPLEMENTATION

1. Database Security:

a. Database tables are designed in such a way that it doesn't allow invalid or empty values for a given column.

   For example – *USERBASE* table which stores user's private data cannot have an empty column for '*userId*' and '*password*' column.

b. Database queries are used with '*PreparedStatement*' methods so as to minimize the risk of SQLInjection.

2. Password Security:

a. At the time of registering, the user needs to enter his password twice so as to avoid accidental invalid passwords.

b. Next the password is hashed using MD5 algorithm after the registration form is successfully submitted, the data now is stored into the database (Application does not store plaintext passwords).

c. Login forms also have a hashing function which converts user entered data into a hash and compare them to the data from the database.

d. Each form goes through a validation check before the submission.

e. ‘*Forgot password utility*’ also has validation checks to only authorize the original owner of the application to change his/her password.

3. **Exception Handling:**

a. Java classes of the application contain efficient code to handle exceptions or errors.

b. A *web.xml* file has also been defined to hide the errors from the outside world upon deployment of the project, which prevents users with malicious intent, who can use the exception data for gaining access into the application.

Limitations of the Project:

- The application is not yet fully automated. It requires a person to evaluate the submissions and update the results.
- A user can submit the solution to a problem only once in a session.
- While adding new problems to the database, the admin has to use proper html tags to get proper formatting of the problems.

Future Application of the Project:

- A programming language compiler could be added, which could totally eliminate the need of an evaluator for the problems.
- The application could be ported to a framework like '***STRUTS***' for much better security and functionalities.
- The project could be modified to run on a multi-server architecture if the need or number of requests increases substantially.

Bibliography:

- Introduction to Database Systems (***Bipin C Desai***).
- Software Engineering, A Practitioner's Approach (***Roger S. Pressman***).
- Introduction to Software Engineering (IGNOU BCS-051).