

UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

Report on System Engineering Lab Project

Course Code : CSE 321

Course Title : Software Engineering

Submitted by:

Name : Sheikh Nafez Sadnan

Reg. No. : 20101106

Roll No. : 106

Section : B(2)

Submitted To:

Fahad Ahmed

Lecturer

Department of CSE

University of Asia Pacific

Group ID:

B2-G5

Project Name:

Code Samlao.

Motivation:

As an adept programmer we have breezed through a lot of coding tutorial and participated in different contests. There are many popular programming challenge platforms worldwide. Such as,

- **Codeforces** [1]: Codeforces is an online judge sponsored by TON which, hosts contests and has sets of problems for users to solve and practice. Codeforces has a rating system to gauge the performance of a user compared to another user. This site also a dedicated community of programmers who share and discuss their knowledge and expertise on programming.
- **HackerRank** [2]: This is a platform that provides users with programming challenges and contests to improve their skills as a programmer. HackerRank offers a variety of programming languages, domains, and skill levels for users to choose from. HackerRank offers resources to prepare for coding interviews. It is a great website for aspiring programmers looking for a job in the field of Computer Science.

- **CodeChef [3]:** Codechef offers its users opportunity to participate in coding challenges, contests and hackathons. It provides a wide range of problems that are designed to improve the problem-solving skills of the users. It provides two types of membership to a user. Free and Pro. Both types of users can learn from provided contents but pro users have access to quizzes, practical projects, guided video solutions and etc.
- **Codewars [4]:** According to the Codewars website, it helps its users to go from beginner to expert and beyond. The opportunities Codewars provides its users are are:
 - Get new perspectives
 - Learn new languages
 - Compete with peers
 - Build self-confidence
 - Become a mentor
- **SPOJ [5]:** SPOJ or Sphere online judge is also a website that contains thousands of programming challenges for users to solve. It supports over 45 programming languages, and users can solve challenges to hone their coding skills as well as earn ranks and badges.

Due to their popularity, hosting instant contests are difficult and sometimes applicants are put on a waitlist. We wish to create a platform that will work as a solution for this issue and provide opportunities to the people that urgently needs it.

Imagine a teacher who is need to host a contest often to evaluate his/her students' progress or increase their enthusiasm in programming. Our platform will provide them with instant and easy access whilst fulfilling the requirement.

Problem Statement:

We will be building a platform for programming contest. This platform will contain features such as:

- **Hosting contests:** Contests are collection of problems that participants can solve in order to gain points/scores. Teachers or professionals will be able to host contests in order to judge their students or colleagues.
- **Participating in programming contests:** A contest needs participant. Once any user or organization hosts a contest, selected participant will be eligible to take part in it through our website.
- **Practice programming challenges:** Users will be able to practice challenges and hone their skill without having to participate in contests. This will help refine his/her skills and boost confidence. This will encourage users to participate in contests more and sharpen own skills through competition.

Objectives:

Our objective is to provide an online platform for enthusiastic programmers to test their skills by solving challenges and participate in contests to prove their proficiency.

Project output:

Our primary target is the nurturing of aspiring programmers. Here are some of the project outputs for our project that should help them:

- ❖ An online platform
- ❖ Problem management System
 - Contest creation and management
 - Problem submission and evaluation system
 - Leaderboard system
 - Analytic system
- ❖ User Guide
 - Notification
 - User friendly UI
 - Documentation and support

Project Feasibility Analysis:

What if the system wasn't implemented?

Ans: There is lack of local programming judge websites. "*Code Samlao*" is an interactive programming judge website that will encourage students, developers, coders and programming enthusiasts to practice and develop their skills. They will go through different types of problem solving and participation in time-based programming contests.

So, If the "*Code Samlao*" programming judge website is not implemented, then user will miss a local-convenient programming judge website. This could lead to a lack of opportunity for individuals who are looking to enhance their abilities and skills and it may hinder their opportunity to compete in the competitive job market. Because, many companies and IT farms calls programmers to their interview by noticing the individual's performance in different competitive programming judge websites.

What are current process problems?

Ans: The current process following in "*Code Samlao*" can lead to some limitations and lack of abilities like-

1. Insufficient budget, human resource and time schedule to add additional features like:
 - User feedback and support for users,
 - Programming related QnA forum,

- Articles about different programming concepts (Data Structures, Algorithms, Discrete Mathematics, Object-oriented programming etc.) ,
 - Programming related video-lectures like in YouTube,
 - Programming language tutorials (C, C++, Java, Python, PHP, Ruby, Swift, Perl etc.)
 - Dedicated android/iOS app of “*Code Samlao*” (It would help to reach more users)
2. Database maintenance issue.
 3. Insufficient exercise problems for practice purpose.

How will the proposed system help?

Ans: The proposed system of “*Code Samlao*” will please our users with different facilities like- participating in different time-bound competitive programming contests and develop their programming knowledge by solving challenges, also test their skills. Top tier positions can receive rewards (If contest authority permits) Here, participants can practice and took part in different types of programming contests, they would introduce to verify programmers around the world. It may also help them to become shortlisted for top tier software giants for interview calls, according to their performance in the contests and solving records.

Is new technology needed? What skills?

Ans: No. so far, no new technology (such as- Artificial intelligence, Machine learning, Block chain, Edge Computing, IoT etc.) concept is not required for the “*Code Samlao*” programming judge website.

The developer team has used HTML, CSS, Bootstrap and JavaScript for the front-end; Python & Django for the back-end; MySQL for the database. Also, IDE's like PyCharm and Visual Studio Code for editing. Also, it requires server related tool like "XAMPP" with at least Apache, MySQL & phpMyAdmin enabled. It requires testing purpose tools like "Selenium".

But for users, it may require knowledge of different classical programming languages (C, C++ etc.), database knowledge (e.g. MySQL) and problem-solving skills.

What facilities must be supported by the system?

Ans: The facilities that must be supported by the "*Code Samlao*" system may include-

1. User-friendly interface.
2. Proper authentication system for users.
3. Programming contests attired with different level/topics of challenges with reward points.
4. Problem solving output verdict must be accurate.
(Accepted/Failed/Environmental Error)
5. Facility for users to track their progress.
6. Strict time management for the contests.

What is the risk associated with cost and schedule?

Ans: The risks associated with cost and schedule in "*Code Samlao*" programming judge website is-

1. Late in development process resulted in delay in delivery of the whole system.
2. Unexpected expenses like requirement of buying new gadgets/devices/software for the system and the development process doesn't go in smooth.
3. Server cost & server management complexity effects on schedule.
4. Different bugs arise in development process.
5. Functional issues in testing and debugging.

Are there legal issues?

"Code Samlao" programming judge website takes user privacy concern very seriously. It does not promote or condone any illegal or harmful activities; it does not promote hatred against any cast/group/occupation/gender/religion/nation/political party or any living or dead person. Problem/challenge scenario portrayed in the website has no intention to spread hatred.

"Code Samlao" only collect the minimum amount of information necessary to provide the user experience butter-smooth and doesn't track or sell user's personal information.

For the user's side, violation of terms and conditions of "Code Samlao" may resulted in termination of user account. Also, in extreme cases (like cyber bullying, threads or harassment created by the individuals) can lead to legal actions.

Technical Requirement:

We have decided to opt in agile methodology. This methodology refers to breaking down the project into small, manageable tasks and delivering working software in iterative sprints. We determined it will be well suited for our system development since it allows flexibility and encourage close collaboration between developers and stakeholders.

Here is the design pattern of the methodology we decided to follow:

Phase	Name	Activity	Time
Module 1	Pre-production	1. Forming the development team. 2. Discussion on the concept. 3. Deciding on a preliminary design for the project.	Week 1
Module 2	Resource Gathering	1. Gathering requirements. 2. Collecting problem sets. 3. Rearranging development tools.	Week 1-2
Module 3	User Profile	1. Creating “User Profile”. 2. Option to “Edit Profile”.	Week 2-3

		3. Option to “Set or Change User Avatar”. 4. Feedback Collection.	
Module 4	Level 1 for project “Code Samlao”	1.UI Design. 2.Authorization. 3.Contest Management. 4.Problem Integration. 5.Test Case Generation. 6.Compiler Integration. 7.Alpha Testing. 8. Feedback Collection.	Week 3-5
Module 5	Level 2 for project “Code Samlao”	1. UI redesign. 2. Github repository checking. 3. Database creation. 4. Compiler Integration 5. Alpha Testing 6. Feedback Collection.	Week 4-5
Module 6	Level 3 for project “Code Samlao”	1.UI redesign 2. Database connection	Week 5-7

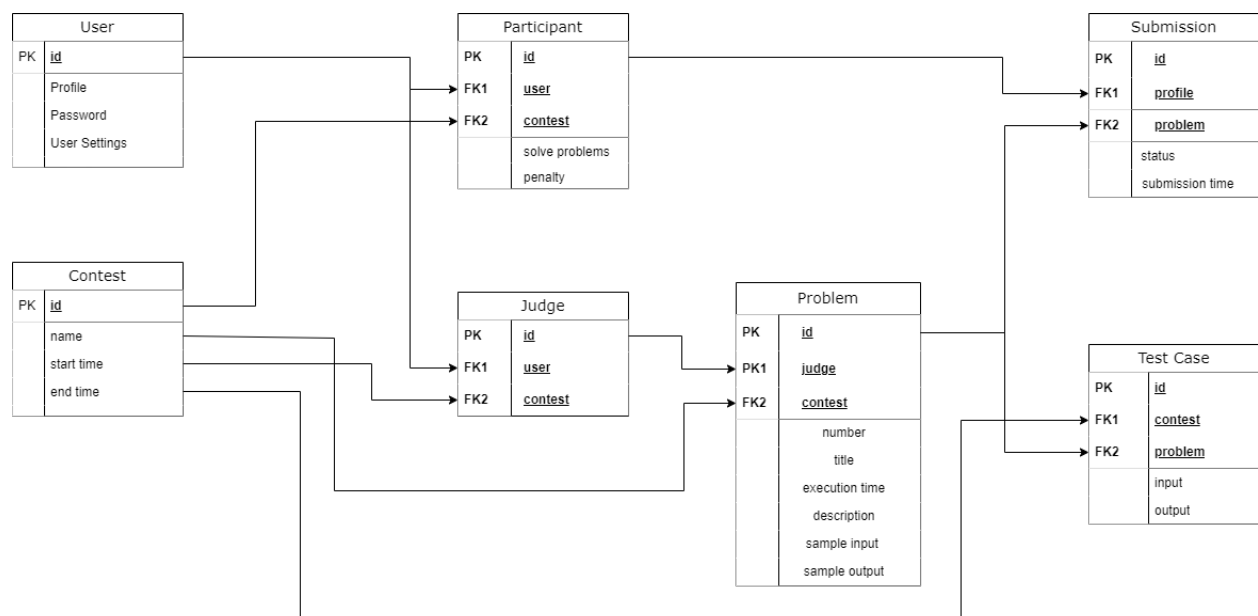
		establishment. 3.Sign up confirmation 4. Contest Management. 5. Problem Integration. 6.Scoreboard implementation. 7.Alpha Testing. 8.Feedback Collection.	
Module 7	Level 4 for project “Code Samlao”	1.UI Design. 2.Project app design. 3.Judge panel design. 4.Problem Integration 5.Test Case Generation 6.Compiler Integration 7.Alpha Testing 8.Feedback collection.	Week 7-8
Module 8	Level 5 for project “Code Samlao”	1.UI Design. 2.Database management. 3.Contest Management. 4.Managing leaderboard.	Week 8-10

		5.Compiler Integration 6.Submission verification. 7.Alpha Testing 8.Feedback Collection	
Module 9	Final Test Phase	1. Beta testing. 2. Addressing any issues and bugs. 3. Feedback Collection.	Week 10
Module 10	Product Release	1. Website Publishing. 2. Monitor Server performance. 3. Feedback Collection. 4. Address any interference.	Week 11
Module 11	Future Support	1. Implement bug fixes and new features. 2. Collect bug reports and feedbacks.	Continuous

Methodology

Here are different diagrams showing our approach to the system and its operation:

ER Diagram:



Our web system contains 7 entities.

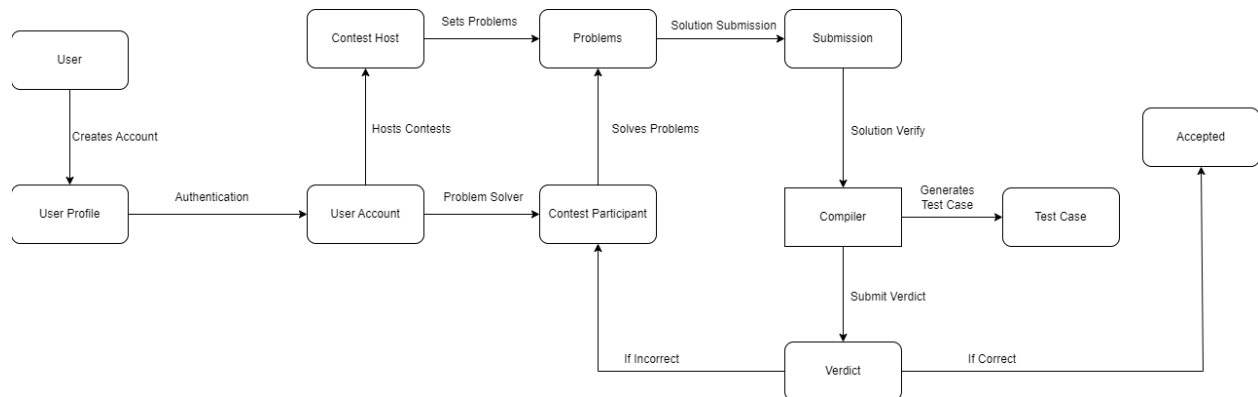
- **User:** Users are the ones who will interact with the system. There will be three types of users:
 - Participant
 - Judge
 - System Developers/Admin

Each user will have a specific id that will be used as primary key in the database. They will also have a profile which they will be able to

edit using User Settings. They will have to use password to secure their account.

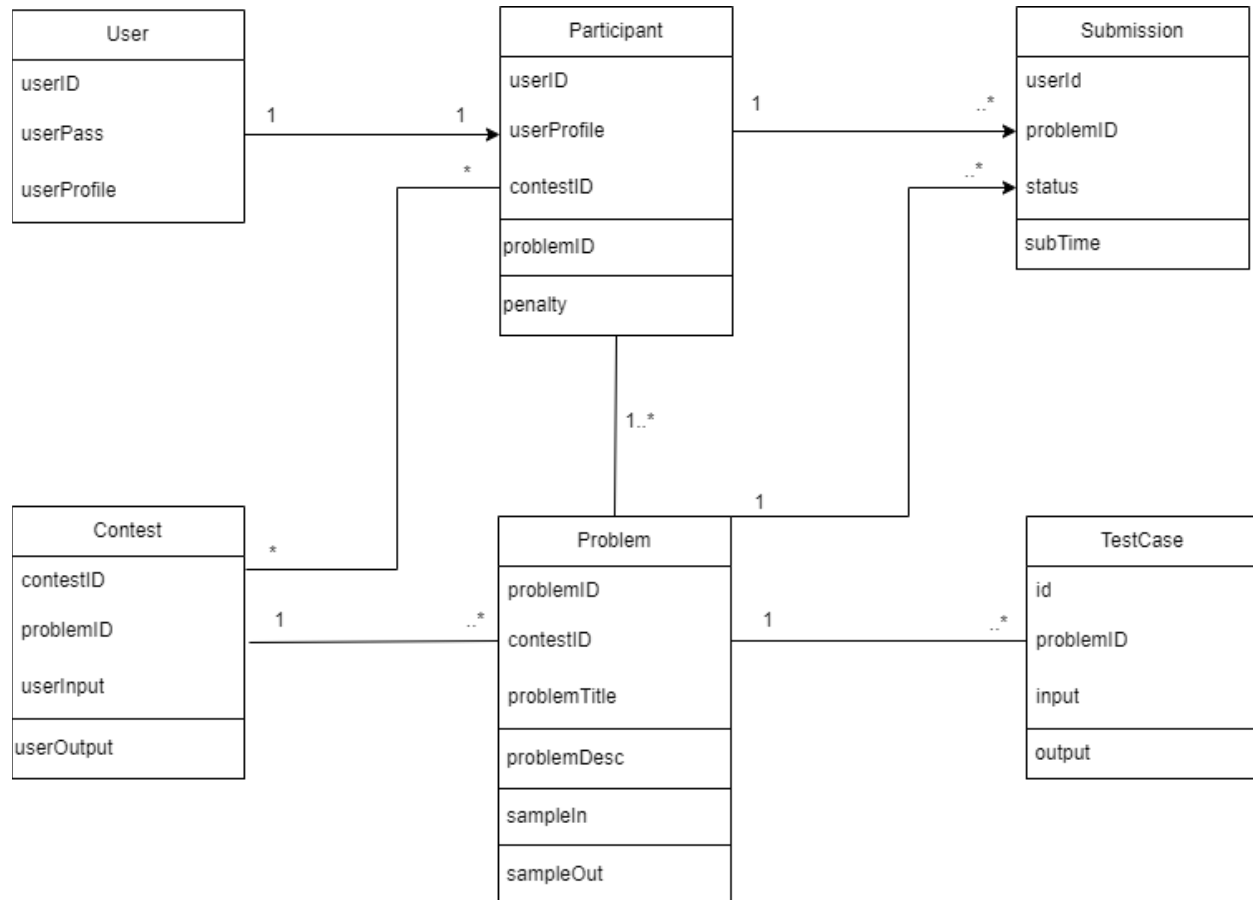
- **Participant:** Participants are a type of users. They will take part in contests and solve problem. They will also have a special id to be used as primary key. In the participant class, they will have their user id and contest id as foreign key.
- **Contest:** Contest entity will have designated id assigned to them. The id will be used as a primary key. It will also have entity such as name, start time and end time.
- **Judge:** Judges are the type of users who will be setting problems for participants to solve. They will have id as their primary key. And have user id and contest id as foreign key.
- **Problem:** Each contest will have problems assigned by judges. Here the primary keys are problem id and foreign keys are judge id and contest id. Each problem will have:
 - Specific number
 - Title
 - Execution time
 - Description
 - Sample input and
 - Sample output.
- **Test Cases:** Test Cases will have designated id as their primary keys and contest id and problem id as their foreign key. Test cases are connected to input and output.
- **Submission:** Each submission will have specific id for their primary key. Submission will have participant id and contest id as foreign key.

DFD Diagram:



This Data Flow Diagram illustrates how the interaction between and system will happen. Initially a user creates an account providing required information which generates a personal user profile. Once the authentication procedure is complete the user will have an account to himself. Contest hosts/judges will set problems which contest participants will solve. The problem solves submitted by the participants will be verified using the solution provided by the judge. Given solutions will be checked by running test cases using compiler. Then the verdict will be generated. If the verdict is incorrect the participants will be informed and if it is correct the submission will be accepted.

UML Diagram:



This diagram represents our programming contest judge web system. There are users with profile, id and pass acting as participant. A user can act as only one participant in a contest. Participants are identified specific features such as userID, userProfile, contestID, problemID. Penalty is given to the participant for failed submission. One participant can do multiple submission. Also, there can be many contests and many participants.

Participants will be solving problem which will be verified by generating TestCase. There can be one or many test cases for a single problem.

Software Process Model:

Our followed model is agile methodology. Here is how we divided our necessary tasks into sprints:

Module 1: Pre-production

➤ Sprint

- ⇒ Forming a development team.
- ⇒ Proposing concept for the system.
- ⇒ Designing a preliminary function for the project.
- ⇒ Selecting requirements.

➤ Time

⇒ Week 1

Steps	Activity
Analysis	<ul style="list-style-type: none">• Project process discussion.• Finding out approximate cost.• Finding out approximate time required.
Design	<ul style="list-style-type: none">• Making a temporary diagram for the whole system.

Code	-
Test	-
Feedback	-

➤ Backlog

⇒ Different types of diagrams containing the project plan.

Module 2: Resource Gathering

➤ Sprint

- ⇒ Gathering requirements.
- ⇒ Collecting problem sets.
- ⇒ Collecting developments tools.
- ⇒ Rearranging developments tools.

➤ Time

⇒ **Week 1 – Week 2**

Steps	Activity
Analysis	<ul style="list-style-type: none"> • Figuring out tools to use.
Design	<ul style="list-style-type: none"> • Installing required tools. • Github repository setup.
Code	-

Test	-
Feedback	-

➤ Backlog

- ⇒ Github repository for online workspace.
- ⇒ Coding environment for every contributor.

Module 3: User Profile

➤ Sprint

- ⇒ Create “User Name”.
- ⇒ Create “Password”.
- ⇒ Create “User Avatar”.
- ⇒ Create “Profile”.
- ⇒ Feedback Collection.

➤ Time

- ⇒ **Week 2 – Week 3**

Steps	Activity
Analysis	<ul style="list-style-type: none"> • Contents to be in a user profile. • How will user sign in and log out interaction play out.

Design	<ul style="list-style-type: none"> • HTML design for create account and home page. • Opening new database.
Code	<ul style="list-style-type: none"> • Establishing CSS and HTML. • Connecting views.py with html.
Test	<ul style="list-style-type: none"> • Alpha testing done.
Feedback	<ul style="list-style-type: none"> • UI is too bland.

➤ Backlog

- ⇒ Temporary overview of the whole system.
- ⇒ Authentication system for the users.

Module 4: Level-1 of “Code Samlao”

➤ Sprint

- ⇒ UI design
- ⇒ Authorization
- ⇒ Contest system design.
- ⇒ Problem Integration
- ⇒ Test Case Generation
- ⇒ Compiler Integration
- ⇒ Alpha Testing

⇒ Feedback Collection.

➤ Feedback Implementation

⇒ Updated CSS.

⇒ Updated HTML.

➤ Time

⇒ Week 3 - Week 5

Steps	Activity
Analysis	<ul style="list-style-type: none">• Finding out all the apps to sort the project.• Separating apps by their purpose
Design	<ul style="list-style-type: none">• Setting up views.py for each app.• Configuring logic in each views.py.
Code	<ul style="list-style-type: none">• Installing apps in the project folder.
Test	<ul style="list-style-type: none">• Alpha testing done.
Feedback	<ul style="list-style-type: none">• Issue setting up the project from github.

➤ **Backlog**

- ⇒ Authentication system with proper UI.
- ⇒ Interactable system.

Module 5: Level-2 of “Code Samlao”

➤ **Sprint**

- ⇒ UI redesign
- ⇒ Github repository checking.
- ⇒ Database creation.
- ⇒ Compiler Integration
- ⇒ Alpha Testing
- ⇒ Feedback Collection.

➤ **Feedback Implementation**

- ⇒ Updating setup instruction in readme.

➤ **Time**

- ⇒ **Week 4 – Week 6.**

Steps	Activity
Analysis	<ul style="list-style-type: none"> • Assigning subsystem of the whole system to project members. • Github instructions unclear.
Design	<ul style="list-style-type: none"> • Updating CSS and HTML.
Code	<ul style="list-style-type: none"> • Forking projects into personal repository. • Configuring views.py in the apps folder.
Test	<ul style="list-style-type: none"> • Alpha testing done.
Feedback	<ul style="list-style-type: none"> • Database error after logging in.

➤ Backlog

- ⇒ Platform to store data.
- ⇒ System able to collect data.

Module 6: Level-3 of “Code Samlao”

➤ Sprint

- ⇒ UI redesign
- ⇒ Database connection establishment.
- ⇒ Sign up confirmation

- ⇒ Contest Management.
- ⇒ Problem Integration.
- ⇒ Scoreboard implementation.
- ⇒ Alpha Testing.
- ⇒ Feedback Collection.

➤ Feedback Implementation

- ⇒ Setting up xampp.
- ⇒ Setting up apache server.
- ⇒ Updating database.

➤ Time

⇒ Week 5 – Week 7

Steps	Activity
Analysis	<ul style="list-style-type: none"> • SQL implementation analysis. • Database connection with system.
Design	<ul style="list-style-type: none"> • Sign up page html update. • CSS updated.
Code	<ul style="list-style-type: none"> • SQL for data table changed. • Model.py connection with database.

Test	<ul style="list-style-type: none"> • Alpha testing done.
Feedback	<ul style="list-style-type: none"> • Score board isn't updating. • Contestants are not receiving points.

➤ Backlog

- ⇒ Viewable live scoreboard.
- ⇒ Leaderboard system.
- ⇒ Updated UI with gradient colors.

Module 7: Level-4 of “Code Samlao”

➤ Sprint

- ⇒ UI Design
- ⇒ Project app design.
- ⇒ Judge panel design.
- ⇒ Problem Integration
- ⇒ Test Case Generation
- ⇒ Compiler Integration
- ⇒ Alpha Testing
- ⇒ Feedback collection.

➤ Feedback Implementation

- ⇒ Reestablishing scoreboard connection.
- ⇒ Resolving issues in model.py
- ⇒ Updating database

➤ Time

⇒ Week 7 – Week 8

Steps	Activity
Analysis	<ul style="list-style-type: none">• Testing solution through compiler.• Discussion about problem submission.
Design	<ul style="list-style-type: none">• Updating HTML and CSS.
Code	<ul style="list-style-type: none">• Configuring views.py• Configuring urls.py• Configuring models.py
Test	<ul style="list-style-type: none">• Alpha testing done
Feedback	<ul style="list-style-type: none">• Submission Error• Password confirmation email not being sent.

➤ Backlog

- ⇒ UI for uploading problem.
- ⇒ UI for submission page.
- ⇒ Interactable contest system.

Module 8: Level-5 of “Code Samlao”

➤ Sprint

- ⇒ UI Design
- ⇒ Database management.
- ⇒ Contest Management.
- ⇒ Managing leaderboard.
- ⇒ Compiler Integration
- ⇒ Submission verification.
- ⇒ Alpha Testing
- ⇒ Feedback Collection

➤ Feedback Implementation

- ⇒ Fixing database connection.
- ⇒ Fixed verdict generation for submission.
- ⇒ Updating database.
- ⇒ Solving issue in app folder.

➤ Time

⇒ **Week 8 – Week 10**

Steps	Activity
Analysis	<ul style="list-style-type: none">• Authentication system.• Problem submission verdict.• Profile management.

Design	<ul style="list-style-type: none"> • Updating system overview.
Code	<ul style="list-style-type: none"> • Merging submissions. • Changing CSS and HTML. • HTML and views.py connection. • Change in models.py.
Test	<ul style="list-style-type: none"> • Alpha testing done.
Feedback	<ul style="list-style-type: none"> • No error detected.

➤ Backlog

- ⇒ Live verdict of submission.
- ⇒ Live score update.
- ⇒ Improved UI.
- ⇒ User profile options.

Module 9: Final Test Phase

➤ Sprint

- ⇒ Beta Testing
- ⇒ Address any issues and bugs.
- ⇒ Feedback Collection.

➤ Time

- ⇒ **Week 10**

Steps	Activity
Analysis	<ul style="list-style-type: none"> • Contest results. • Point table update.
Design	<ul style="list-style-type: none"> • Updating CSS and HTML.
Code	-
Test	<ul style="list-style-type: none"> • Beta testing done.
Feedback	<ul style="list-style-type: none"> • No error detected.

➤ Backlog

⇒ Final project.

Module 10: Product Release

➤ Sprint

- ⇒ Website Publishing.
- ⇒ Monitor Server Performance.
- ⇒ Address bugs and reports.
- ⇒ Feedback Collection.

➤ Feedback implementation

- ⇒ Resolving issue in view.py logic.
- ⇒ HTML updated.

➤ Time

⇒ Week 11

Steps	Activity
Analysis	<ul style="list-style-type: none">• Website performance.• User interaction.
Design	-
Code	-
Test	<ul style="list-style-type: none">• Beta testing done.
Feedback	<ul style="list-style-type: none">• Server became overloaded.

➤ Backlog

⇒ Active website.

⇒ Active server.

Module 11: Future Support

➤ Sprint

⇒ Implement Bug Fixes.

⇒ Implement New Features.

⇒ Collect bug reports.

⇒ Feedback collection.

➤ Feedback implementation

⇒ Contacting server management.

Steps	Activity
Analysis	<ul style="list-style-type: none">• System performance checkup.
Design	-
Code	-
Test	-
Feedback	<ul style="list-style-type: none">• Continuous.

➤ Backlog

⇒ Useable system.

⇒ Working computer programming contest judge.

Sample Code:

```
from django.shortcuts import render,HttpResponse
import subprocess,os
from django.contrib.auth.decorators import login_required
from django.conf import settings
import multiprocessing as multi
import time
import threading
```


Project Development Resource:

This includes the list of developing resources used in the system development.

- ❖ Django Documentation.
- ❖ MySQL API.
- ❖ TinyMCE API.
- ❖ “Multiprocessing” Package for python.
- ❖ “Subprocess” module for python.

This is a sample for our compiler that takes submitted codes and generates verdict.

```
from django.shortcuts import render,HttpResponse  
import subprocess,os  
from django.contrib.auth.decorators import login_required  
from django.conf import settings  
import multiprocessing as multi  
import time  
import threading  
from contest.models import TestCase  
# Create your views here.
```

```
def codecheck(send,input,file_name):
```

```
    Input=input
```

```
    p = subprocess.Popen([file_name], stdin=subprocess.PIPE,  
stdout=subprocess.PIPE)
```

```
    stdout, _ = p.communicate(input=Input)
```

```
    send.send(stdout)
```

```
@login_required
```

```
def runcode(request):
```

```
    if request.method=='POST':
```

```
        problemid=request.POST['pk']
```

```
        code=request.POST['code']
```

```
        file_name=os.path.join(settings.MEDIA_ROOT_CODE,  
request.user.username)
```

```
        file=str(file_name+".cpp")
```

```
        #insert the code into to file
```

```
        with open(file, "w") as f:
```

```
            f.write("".join(code))
```

```
        #get testcase
```

```
        test_casees=TestCase.objects.filter(problems=problemid)
```

```

try:
    submissionstate=""
    process=subprocess.Popen(['g++',file],stderr=subprocess.PIPE)
    _,stderr=process.communicate()
    if "error" in stderr.decode():
        submissionstate="compilar error"
        print("compilar error")
    else:
        subprocess.run(["g++", "-o",file_name , file])

    recv, send = multi.Pipe(False)
    for testcase in test_casees:

        output=testcase.output.read()
        input=testcase.input.read()

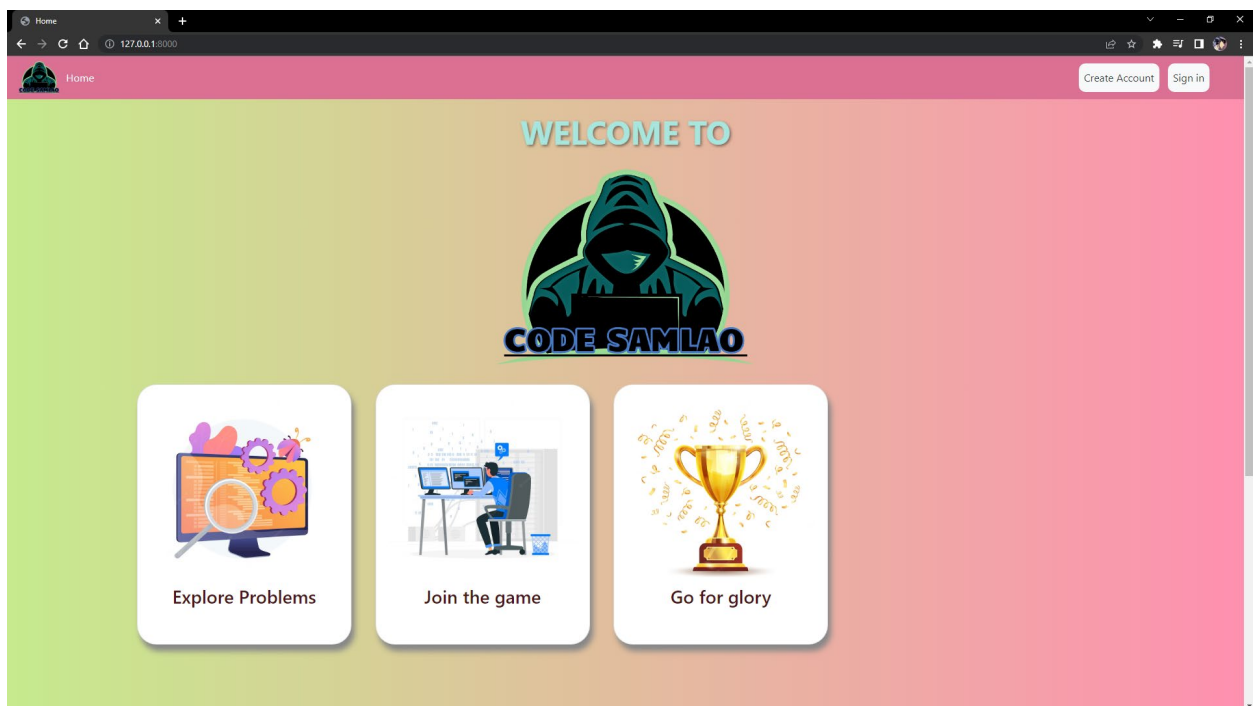
        p = threading.Thread(target=codecheck,
args=(send,input,file_name))
        p.start()
        runtime=float(testcase.problems.execution_time)
        time.sleep(runtime)
        ret = recv.poll()
        if ret == False:
            submissionstate='TLE'
            print("TLE")
            p.kill()

```

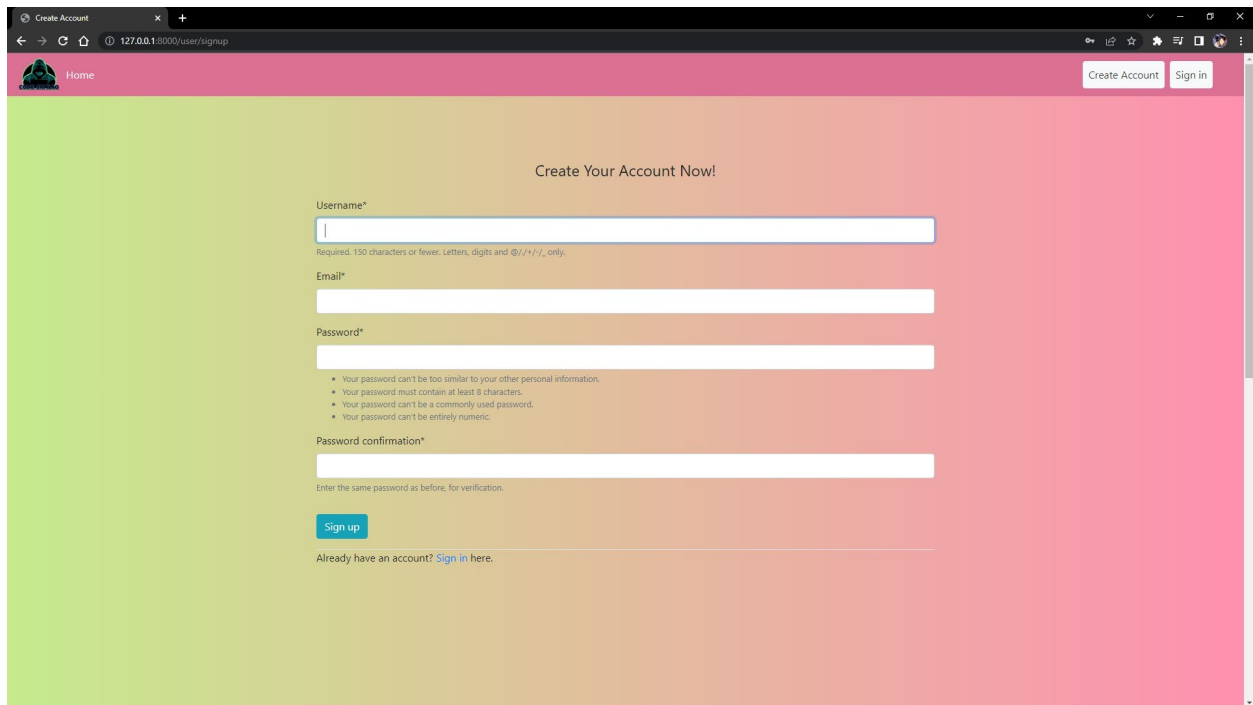
```
        break
    else :
        recive=recv.recv()
        if recive!=output:
            print(recive)
            submissionstate="Wrong"
            print("Wrong")
            break
    else:
        submissionstate="Accepted"
        print("Accepted")
    p.close()
    send.close()
    recv.close()
except:
    None
return render(request, "submission.html",{ 'submissionstate':submissionstate})
```

Final Result of our project:

Homepage:



Signing up to create a new account:



Create Account

Home

Create Account Sign in

Create Your Account Now!

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

[Sign up](#)

Already have an account? [Sign in here.](#)

Log in with account credentials:

Login

Username

E511

Password

☒ Remember me

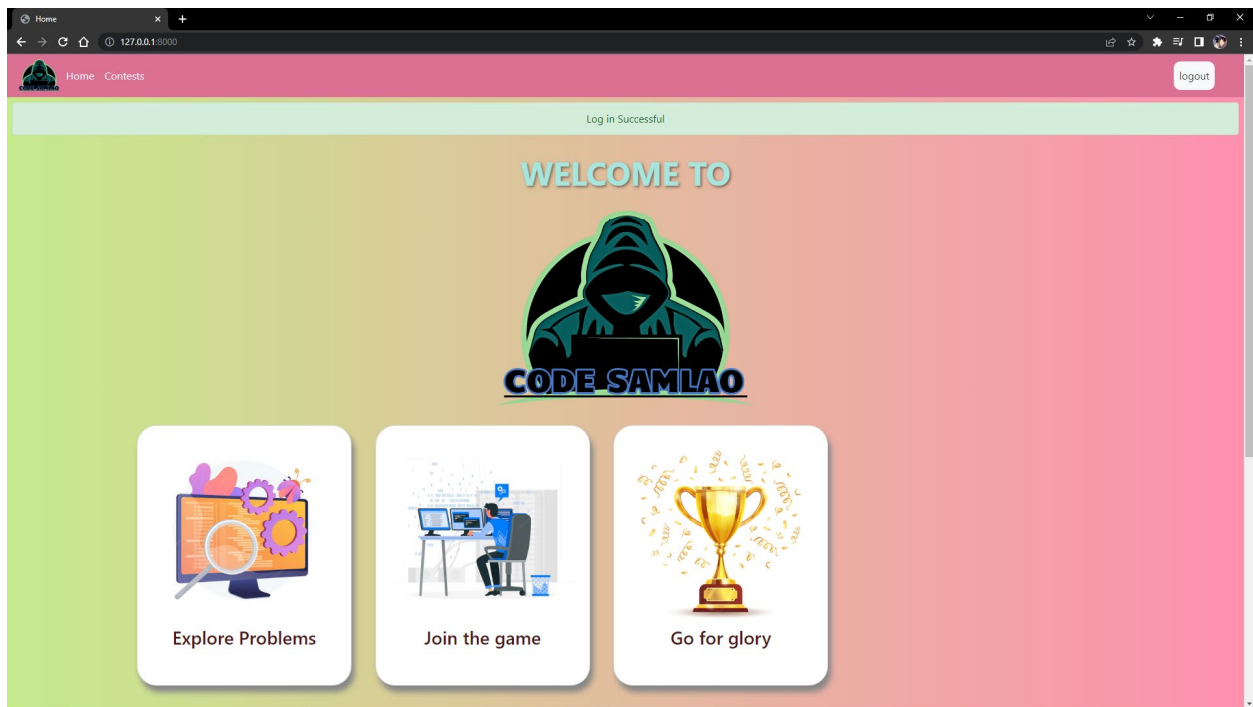
Sign in

[Forgot Password?](#)

Don't have an account?

[Create an account](#)

Home page after first time logging in:



Contest page:

Contests

Home Contests

logout

Running Contests

Start Time	Contest Name	Ending Time	
2023-05-14(Sun) 18:00	Coder Regular Contest 160	2023-05-14(Sun) 19:00	Participant

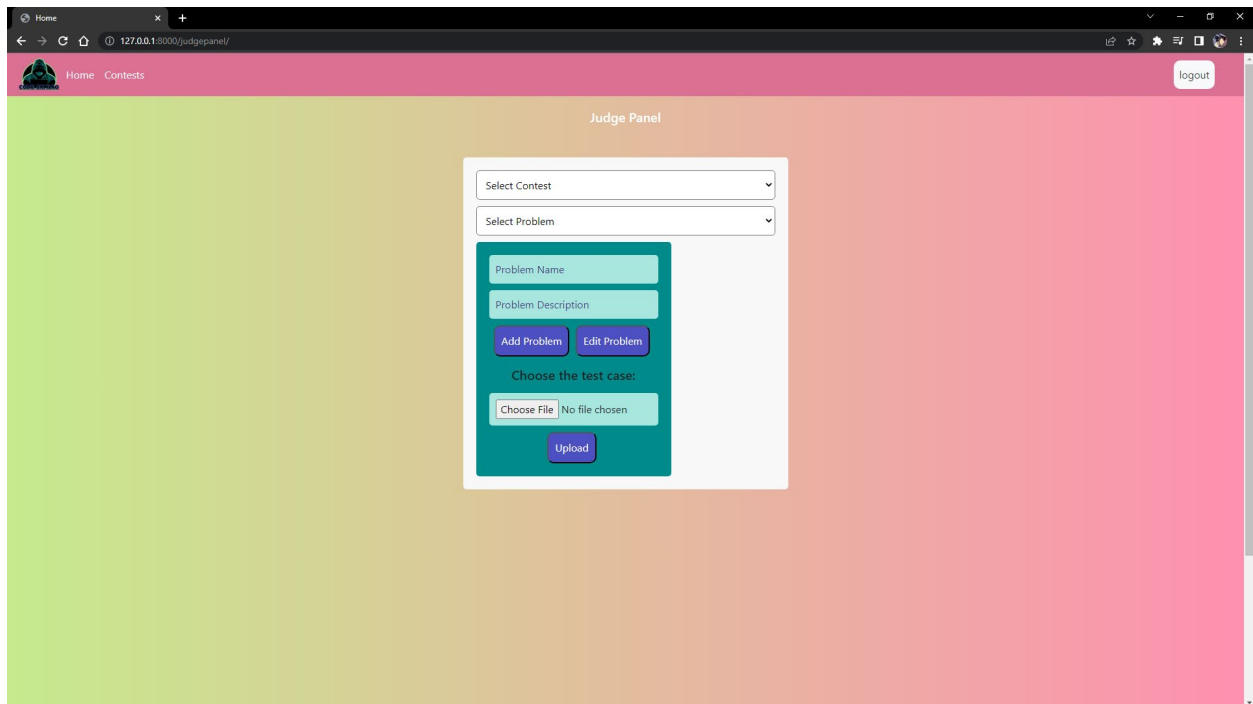
Up Coming Contests

Start Time (local time)	Contest Name	Ending Time	
2023-05-14(Sun) 18:00	Coder Regular Contest 161	2023-05-14(Sun) 19:00	Registration
2023-05-21(Sun) 18:00	Coder Grand Contest 162	2023-05-21(Sun) 19:00	Registration

Previous Contests

Start Time (local time)	Contest Name	Ending Time	
2023-05-14(Sun) 18:00	Coder Regular Contest 159	2023-05-14(Sun) 19:00	End Content
2023-05-21(Sun) 18:00	Coder Grand Contest 158	2023-05-21(Sun) 19:00	End Content

Judge panel for uploading and editing problems:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/judgepanel/". The page has a pink header bar with a "Home" link, a "Contests" link, and a "logout" button. The main content area has a light green background with a "Judge Panel" title. A central form is used for managing problems. It includes two dropdown menus for "Select Contest" and "Select Problem". Below these are input fields for "Problem Name" and "Problem Description". There are two buttons, "Add Problem" and "Edit Problem", positioned below the description field. A section titled "Choose the test case:" contains a file selection interface with a "Choose File" button, a "No file chosen" status, and an "Upload" button.

Home Contests logout

Judge Panel

Select Contest

Select Problem

Problem Name

Problem Description

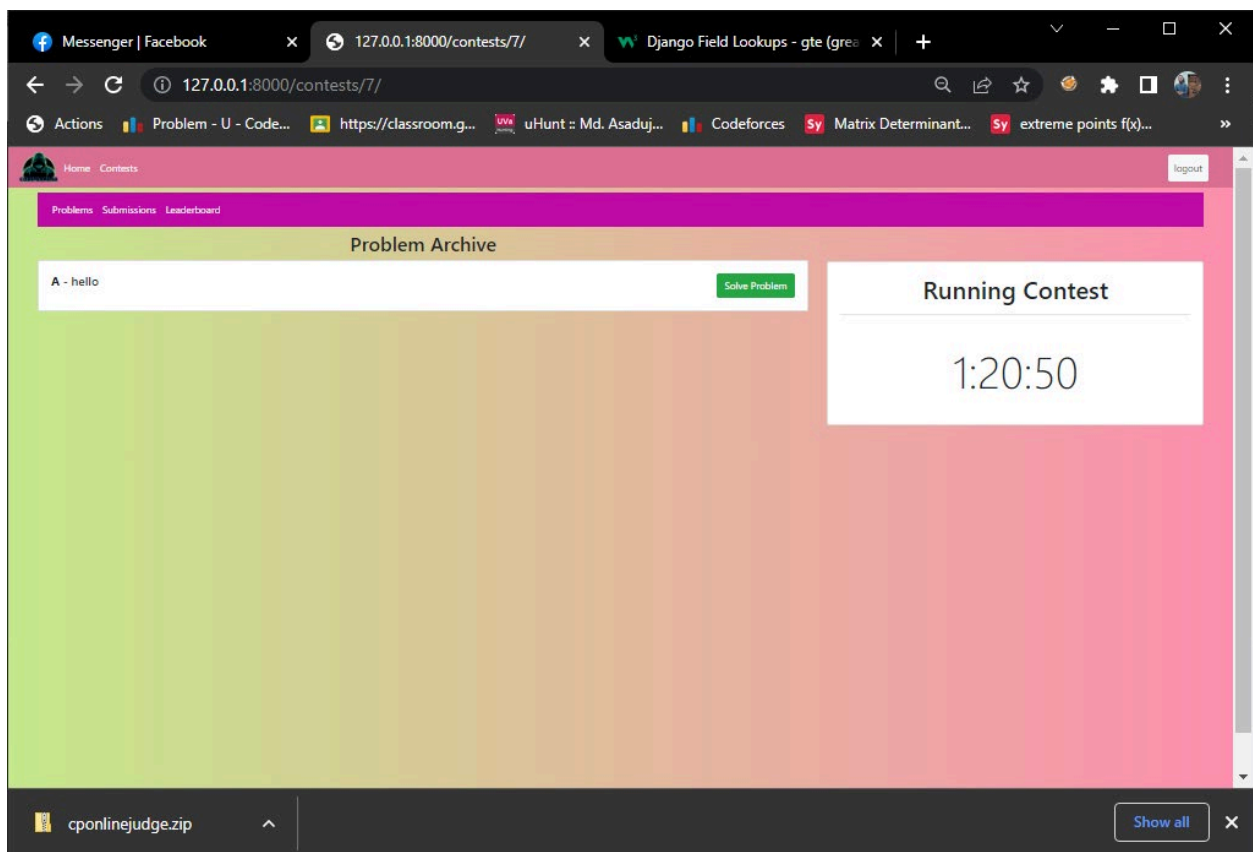
Add Problem Edit Problem

Choose the test case:

Choose File No file chosen

Upload

Running Contest:



Problem and Code submission page:

Messenger | Facebook x 127.0.0.1:8000/contests/7/A/ x Django Field Lookups - gte (gre x +

127.0.0.1:8000/contests/7/A/

Actions Problem - U - Code... https://classroom.g... uHunt :: Md. Asadj... Codeforces Sy Matrix Determinant... Sy extreme points f(x)...

Home Contests logout

Problems Submissions Leaderboard

Problem Description:

hello

My friend give me a number and sad that to give him all number between 1 to this number. Unfortunately, my lab final is coming soon that's why I would not be able to do this. can you help me to get the solution?

Now you have to write a code which is give me the number.

Input
The input will contain an integer N ($1 \leq N \leq 100$).

Output
Print the all number from 1 to N

Sample
Input:
5
Output:
1 2 3 4 5

Author: Noor
Time: 1.00 s
status: Accepted

Running Contest

1:21:23

Coding Area:
Language: C/C++

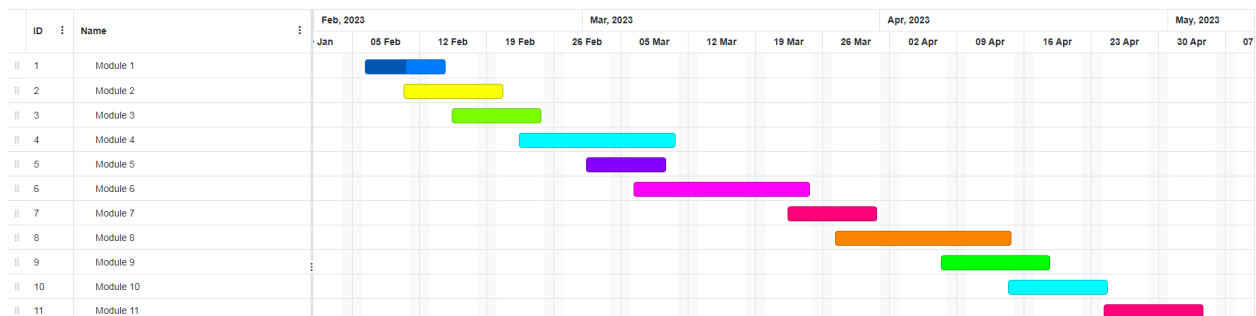
cponlinejudge.zip Show all x

Project Management:

Project Timeline

Our project started at February 05, 2023 and development ended at April 23, 2023.

Each module lasted a minimum of 1 week. Some modules were 2-3 weeks long. Our tasks were divided among contributors and some modules continued parallelly. The project development lasted for 11 weeks. After the development, the continuous support module kicked in. Total work process was separated in 10 modules. And the final module 11, is for continuous future support. Here is the grant chart illustrating our project timeline.



Finance Management:

Here is a table illustrating the cost required for the project.

Activity	Time Cost	Estimated expense in Taka
Information gathering	160 Hours	50000
Pre-production setup	170 Hours	50000
Setting up user experience	250 Hours	100000
Database setup	270 Hours	50000
Server management	Project lifetime	200000+
UI development	320 Hours	125000
Documentation	50 Hours	25000
Approximate Project Costs	1220 and more	600000

Conclusion and Future Learning:

Participating in this project helped me learn a lot. It has taught us:

- ❖ The necessity of co-operation.
- ❖ The importance of communication between contributors.
- ❖ Discipline.
- ❖ Importance of time management.
- ❖ Connection between programming languages.

This project assisted us in improving our proficiency. We learned different tactics and implementation of those in the field of programming. Our project influenced us to get in touch with more advanced development tools and also improved our efficiency in solving problems. We learned how important it is to keep up with the latest developments in the programming world by staying up-to-date with new programming languages, frameworks, and tools. By continuously learning and challenging ourselves, we can improve our competitive programming skills and become a more well-rounded programmer.

To conclude, this has been a valuable lesson for all of us. We got the opportunity to increase our depth of knowledge and experience. Our project will immensely help programmers to improve their coding skills and develop their problem-solving capability. Our project features an interactive user-friendly interface which will attract more users. Realtime scoring and live update will assist aspiring programmers by keeping them on their toes.

References

- [1] "Codeforces," [Online]. Available: <https://codeforces.com/>.
- [2] "HackerRank," [Online]. Available: <https://www.hackerrank.com/>.
- [3] "CodeChef," [Online]. Available: <https://www.codechef.com/>.
- [4] "Codewars," [Online]. Available: <https://www.codewars.com/>.
- [5] "SPOJ," [Online]. Available: <https://www.spoj.com/>.