

CPorBit : An Automated Mentor For Competitive Programmers

BY

MD. ERFANUL ISLAM BHUIYAN

ID: 181-15-10777

AND

MD. SHAH HABIBUL IMRAN

ID: 181-15-10627

AND

TANIMA HOSSAIN

ID: 181-15-10733

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

Supervised By

Saiful Islam

Sr. Lecturer

Department of CSE

Daffodil International University

Co-Supervised By

Dr. Sheak Rashed Haider Noori

Associate Professor & Associate Head

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

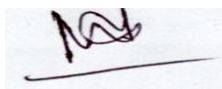
DHAKA, BANGLADESH

JANUARY 2022

APPROVAL

This project, titled “CPOrBit”, was submitted by Md. Erfanul Islam Bhuiyan (181-15-10817) and Md. Shah Habibul Imran (181-15-10627), and Tanima Hossain (181-15-10733) to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation was held on January 2, 2022.

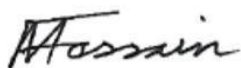
BOARD OF EXAMINERS



Dr. Md. Ismail Jabiullah
Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

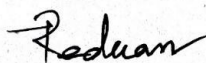
Chairman



Dr. Md. Fokhray Hossain
Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Reduanul Haque
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor

Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Saiful Islam, Sr. Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Saiful Islam
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:



Dr. Sheak Rashed Haider Noori
Associate Professor and Associate Head
Department of Computer Science and Engineering
Daffodil International University

Submitted by:



Md. Erfanul Islam Bhuiyan
ID: 181-15-10777
Department of CSE
Daffodil International University



Md. Shah Habibul Imran
ID: 181-15-10627
Department of CSE
Daffodil International University



Tanima Hossain
ID: 181-15-10733
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing making us possible to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Saiful Islam, Sr. Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in Web Application and Artificial Intelligence to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Supervisor, Saiful Islam, Sr. Lecturer**, for his kind help to finish our project and the Head Of the Department of CSE and other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire coursemate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

With easy access to the internet, we can learn anything with the help of it in this digital era. Also nowadays, web applications add up to that learning process by playing a vital role providing information based on the area of our interest and some extra features that make our experience more enjoyable. Similarly competitive programmers need the help of other experienced people to guide them. And lack of proper guidance or any guidance at all, leads to the downfall of many programmers, specially, the beginners. On the other hand, for competitive programmers improving their rating, performance and polishing it has a huge importance. To do so, one might need to go through that competitive programmers' practice and performance and that kind of guidance is pretty inconvenient to get. Here comes our project. The developed system is a web application aimed at helping competitive programmers to enhance their performance. It will provide them with problem suggestions to improve their performance with mentor support. The front-end was implemented using React[15] and Tailwindcss. On the other hand, Python[16], Django, Django-REST Framework[14], PostgreSQL[21] were used in the back-end. To provide more accurate suggestions, AI is used in the back-end. This system makes programmers' lives easier by guiding them even if they are not from the CSE background. Other programmers can guide and mentor programmers as well. This project will help someone who intends to improve their competitive programming performance by themselves.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgments	iii
Abstract	iv
 CHAPTERS	
Chapter 1: Introduction	1-4
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected Outcomes	3
1.5 Report Layout	3
 Chapter 2: Background	 5-7
2.1 Introduction	5
2.2 Related Works	5
2.3 Comparative Studies	6
2.4 Scope of the Problem	6
2.5 Challenges	7
 Chapter 3: Requirement Specification	 8-15
3.1 Business Process Model and Notation	8
3.2 Use Case Diagram	9
3.3 UML Diagram	12
3.4 Requirement Collection and Analysis	13
3.5 Logical Data Model	13
3.6 Design Requirement	15
 Chapter 4: Design Specification	 16-29
4.1 Front-end Design	16

4.2 Back-end Design	28
4.3 Interaction Design and UX	28
4.4 Implementation Requirements	29
Chapter 5: Implementation and Testing	30-34
5.1 Implementation of Database	30
5.2 Implementation of Back-end	31
5.3 Implementation of Front-end Design	31
5.4 Test Implementation	31
5.5 Test Results and Reports	34
Chapter 6: Conclusion and Future Scopes	35-36
6.1 Discussion and Conclusion	35
6.2 Limitations of Our Application	35
6.3 Scopes for Future Developments	36
REFERENCES	37-38
APPENDICES	39

LIST OF TABLES

TABLES	PAGE
Table 3.2.1: Use Case Description for Sign Up	10
Table 3.2.2: Use Case Description for Log In	10
Table 3.2.3: Use Case Description for Update Profile	11
Table 3.2.4: Use Case Description for Request for problem suggestion	11
Table 3.2.5: Use Case Description for Recommend Problem	11
Table 5.4.1: Test Cases	32

LIST OF FIGURES

FIGURES	PAGE
Figure 3.1.1: Business Process Model and Notation of CPorBit	8
Figure 3.2.1: Use case diagram of CPorBit	9
Figure 3.3.1: UML Class Diagram for CPorBit.	12
Figure 3.5.1: ER Diagram of CPorBit	14
Figure 4.1.1: Screenshot of Landing page for CPorBit	16
Figure 4.1.2: Screenshot of Landing Page down scrolled	17
Figure 4.1.3: Screenshot of Registration Page	18
Figure 4.1.4: Screenshot of Login Page for CPorBit	19
Figure 4.1.5: Screenshot of Standing Page for CPorBit	19
Figure 4.1.6: Screenshot of profile Page for CPorBit	20
Figure 4.1.7: Screenshot of own profile page for CPorBit	21
Figure 4.1.8: Screenshot of Notification Page for CPorBit	21
Figure 4.1.9: Screenshot of Account Settings Page for CPorBit	22
Figure 4.1.10: Screenshot of Profile Settings Page for CPorBit	22
Figure 4.1.11: Screenshot of Current Connections Page for CPorBit	23
Figure 4.1.12: Screenshot of Current Pending Requests Page for CPorBit	23
Figure 4.1.13: Screenshot of Current Requests Page for CPorBit	24
Figure 4.1.14: Screenshot of All Problems Page for CPorBit	25
Figure 4.1.15: Screenshot of Problem Page 1	25
Figure 4.1.16: Screenshot of Problem Page 2	26
Figure 4.1.17: Screenshot of Dashboard	27
Figure 4.1.18: Screenshot of Submissions	27
Figure 4.1.19: Screenshot of Dark Theme for CPorBit	28
Figure 5.1.1: Screenshot of Database Tables of CPorBit	30

CHAPTER 01

INTRODUCTION

1.1 Introduction

The journey of competitive programming is very exciting because it doesn't follow a certain rule. For every programmer, the journey and difficulty are different. There are thousands of problems around so many online judges. But what to solve and what is beneficial for a programmer; determining that is very hard. To do that, we have to observe a programmer's solves, strengths, and weaknesses. And Imagine if we have to do that for every programmer manually how much time and effort that might take, not to mention the amount of time it will take as well. Also, a programmer's progress or capability is constantly changing and with changing situations we need to adapt the suggestions for them as well. Even if we did that part, what about the problems that are going to be suggested to the programmer? There are thousands of problems out there and growing, analyzing all of them and their difficulty then suggesting some is almost an impossible task. Also one has to be constantly upgrading their data of those problems as new problems come. On the other hand, the programmer suffering due to lack of guidelines might lose their interest over time. That is why we decided to build this system to analyze the programmer's performances and suggest problems accordingly to polish their strengths and eradicate their weaknesses.

In the project, we will analyze the programmer's strengths and weaknesses based on their submission of solved problems on an online judge, for example the problem category and difficulty they are solving the problems. Then it will analyze the strengths and weaknesses of the programmer. And to improve that, the system will suggest problems that will improve their weaknesses. And in this process, the programmer will be able to cover up almost all the conventional topics with nearly equal strength.

Though the system will generate problem suggestions based on the users' performance, sometimes a little human touch is needed. We know the value of real human guidance. That is why there is a way to connect with other users, adding them

as mentors or mentees. A mentor will provide custom problem suggestions.

Also, they are competitive programmers; they need to feel that adrenaline rush even when they are practicing. So, we are providing them a gamification part, standings. They can compare their practice speed with others, such as ranking based on total problem solved a week, the total point earned a week, how many days they have continuously solved problems, etc.

In short, this system will help us to identify one's strengths and weaknesses based on contest performance. And it will provide personalized suggestions for users to improve their contest programming skills.

1.2 Motivation

Firstly, during our second semester of university, when we started programming, we didn't know how to start or after learning a topic what to do next. We struggled a lot until we got mentors.

Secondly, After getting mentors, we became too dependent on them. But they couldn't always spare time for us as they were busy themselves and we felt lost at those times.

Thirdly, we've seen many people stopping midway because they didn't have mentors and didn't know how to move forward from a certain point. As a result, they started trying out random topics and eventually gave up.

Fourthly, we've also seen people who didn't follow proper guidelines or got poor guidelines from inexperienced people. Which eventually led to them giving up.

Finally, After facing all these circumstances, we felt it would have been better if there were a system that could guide us properly. We searched the Internet for solutions. But we didn't find any such system. So we decided to try to build such a system and started to study and research about it. After completing our initial study, we saw that it was possible to create such a system and started working on it.

1.3 Objectives

In this project, our main objective is to develop a system for programmers which will guide them as a mentor on what problems they should solve and what topics they should practice after one particular topic. Some specific objectives are needed for this

project.

- It's very important to analyze the system. We need to collect data, identify the system requirements and evaluate this data and requirements.
- Next is to design the model for the learning system. This is a web-based application. So, firstly, design the Front-end part for user interaction, and next, implement the Back-end using python[16] and Django-Rest framework[14].
- Next is to create an AI that'll suggest problems based on a user's performance.
- The last part is to test and validate the system to ensure that it runs successfully.

1.4 Expected Outcome

Using Artificial Intelligence this web based project will analyze an individual's profile and create personalized guidelines more accurately. As these guidelines will update from time to time depending on one's contest performance, it will be very precise, up to date and helpful for contest programmers.

- In every learning process, everyone needs a mentor to guide them. This platform will guide them as a mentor who will help them fulfill their need for a mentor.
- In this web application, an auto-guidance system will make programmers' learning process easy and smooth by providing suggestions on what problems to solve next and which topic to practice now.
- In this web application, we can still keep some human touch by using self mentoring or getting another mentor. One can self mentor by shortlist problems for themselves or can add a mentor who can add problems for them.
- It can also be used as a community where one can get help from their desired programmers by adding them as a mentor.

1.5 Report Layout

As our project is a development project, we will describe our project in six chapters. These six chapters are Introduction, Background, Requirement Specification, Design

Specification, Implementation, Testing, and the last chapter is the conclusion and future scope.

In the First Chapter, We just described the total view of our project, including Introduction, Motivation, Objectives, Expected Outcome, and Report Layout.

The Second Chapter, Illustrates information about the background of our project, like related works, Comparative Studies, the Scope of the problem, and what kinds of challenges we were faced with creating the routine.

The Third Chapter, Describes some requirements and functionality of our project. The business process model and use case modeling were described how to work the routine exactly. In addition, requirement collection and design requirements were specified which types of data needed for this project.

In the Fourth Chapter, Design Specification includes Front-end Design, Back-end Design, Interaction Design, and UX and Implementation Requirements. The types of language used in the Front-end and Back-end will also be described in this section.

In the Fifth Chapter, After completing the Front-end and Back-end design, the next part is implementation and testing. How to implement using a database and user interaction with the system are given here. In addition, the testing result and reports are also viewed in this section.

The Sixth Chapter, is the last part of this project, including two sections: Conclusion and Future Scope. The future scope section describes in detail the features that the system can add to the future.

CHAPTER 02

BACKGROUND

2.1 Introduction

This web application will suggest problems based on a user's performance, strength, and weakness over various data structures and algorithms. As competitive programmers, we felt what a competitive programmer needs to improve their skills. They need good mentor support and problems of various topics to solve. Many online judges provide good problems, ranging from easy to advanced, on various topics. But, there are some topics a programmer is good at, and there are some topics (s)he is not strong enough. Programmers need to practice harder to overcome those weaknesses. But it is difficult to find problems on related topics with adequate difficulty. That is why we decided to develop a system that will generate problem suggestions based on a user's performance, strengths, and weaknesses, with the help of Artificial Intelligence. Also, a user can ask other users to mentor. And mentors can suggest problems too. This way, a user will get the right problem suggestion with the help of both AI and human support.

2.2 Related Work

There are many problems suggesting websites, but none actually meet our needs. The closest work is code-drills[1], which suggests problems to a user based on their performance, strength, and weakness. But this site lacks a few things. A user will not be encouraged to solve problems by just recommendations, he/she will lose interest very soon. And, there is no way to compare yourself with others. Also, AI-generated suggestions are not always enough. Sometimes a human touch is needed. That is why we have brought some unique features, gamification, leaderboard, mentoring, etc. A user will get a challenging and fun experience while improving their performance through earning points, achievements, etc. There will be a global leaderboard for users to compare their performance with other users. A user can ask any other user to be a mentor or mentee. And a mentor can provide suggestions to mentees. This way, a

user's experience will be much better than just getting some problem suggestions.

2.3 Comparative Studies

We have developed our project to overcome the failings of other related works, such as mentoring, gamification, leaderboard. So, we have introduced a gamification system. The user will be rewarded with some points after solving a system-generated problem. Also, the user will get a streak bonus if they keep solving the problems regularly. These streak points will be added to the main points. There will be a global leaderboard where the users will be ranked based on their points. This way, a user will find solving problems more interesting and challenging. It'll not be some boring problem recommendation. And through the leaderboard, a user will be able to compare himself with others. Moreover, the user will get mentor suggestions, which will be a lot helpful as only a real programmer can understand what problems to solve to improve programming performance. These are the unique features of our projects that we didn't find anywhere else, and these features will make a competitive programmer's journey a lot more interesting and easier.

2.4 Scope of the Problem

Programmers, especially competitive programmers, need a lot of guidance initially. Since there are a lot of topics to learn and a lot of problems to solve, it's highly likely for a programmer to get confused and lose track. And soon, they might lose interest in programming. So, they need proper suggestions, guidance, and mentoring. Many young programmers ask us for appropriate directions to improve their programming performance. But it is not always easy to provide proper suggestions, and sometimes finding a good mentor is difficult. Keeping this in mind, we have created a system that'll be online and act as a virtual mentor for a programmer. Also, they can find other users online and ask them to be mentors. This way, they can find proper guidance to improve their performance, with a touch of gamification that'll make their progress interesting.

2.5 Challenges

The biggest challenge was to create a system that'll generate problem suggestions for a user based on their strengths and weaknesses. And to figure out a user's strengths and weaknesses, we need to know how they're performing. We also need to choose an online judge that most users use. We have chosen Codeforces[2], as this is used by many users and is the most popular online judge nowadays. It has a huge number of problems, ranging over numerous topics, specialized for competitive programmers to improve their overall programming skills. We had to collect data from the judge for analyzing a user's performance and generate suggestions for the user. We had to study the API provided by the judge and make a system that'll collect data via the API. For suggestions, we've designed an AI-based tool that'll generate problem suggestions, considering a user's current performance, strengths, and weaknesses. Another big challenge was to design a system for mentorship and to handle the mentor to mentee relationship.

CHAPTER 03

REQUIREMENT SPECIFICATIONS

The specific requirements following which the project has been developed, also a part of the planning of the project. Providing some specific requirements which are many types of models like business process model and notation, use case diagram, UML diagram, ERD diagram, etc.

3.1 Business Process Model and Notation

The application launches on the home page. From there we can access multiple features. Some features are exclusive to logged-in users. The project's main feature recommendation system is exclusive to logged-in users. Logged in users can establish connections playing the role of mentor or mentee and can get a generated problem list based on their data by the system.

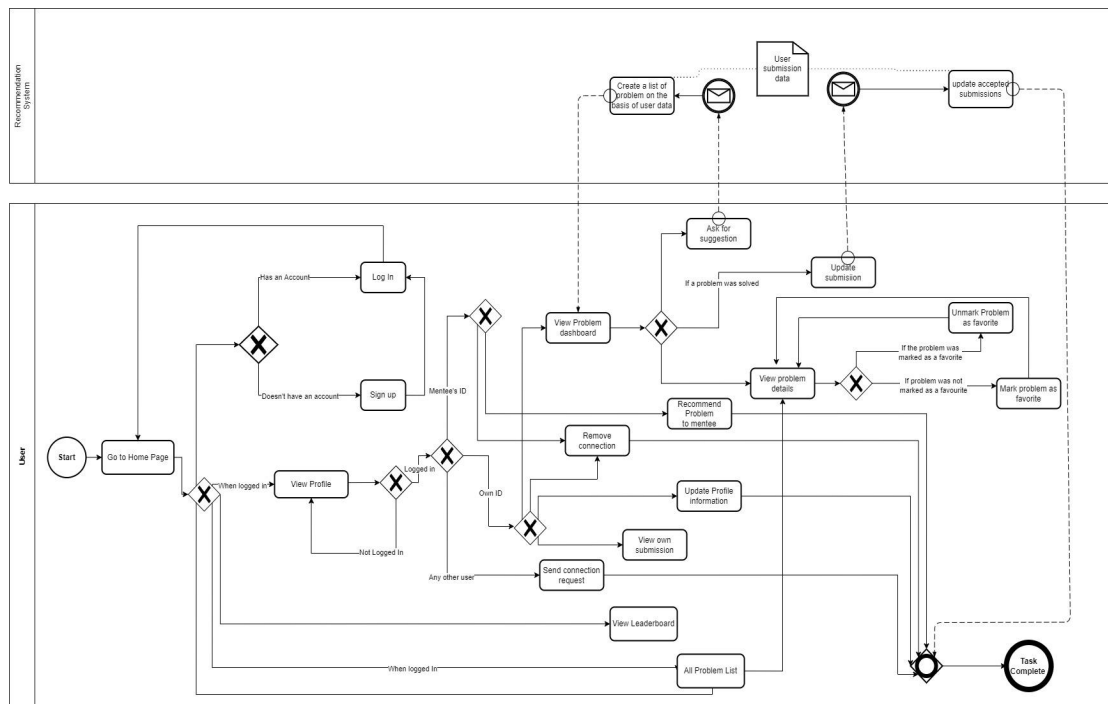


Figure 3.1.1: Business Process Model and Notation of CPorBit

3.2 Use Case Diagram

The use case diagram is undeniably a very functional tool for ensuring preconditions of the project. The use case diagram allows us to have an overview of the whole system. In figure 3.2.1, our use case diagram has two types of users one that can access features that can be accessed by logged in users only defined as mentor/mentee, another is any user which is the users which can access public features and it also has seventeen use cases

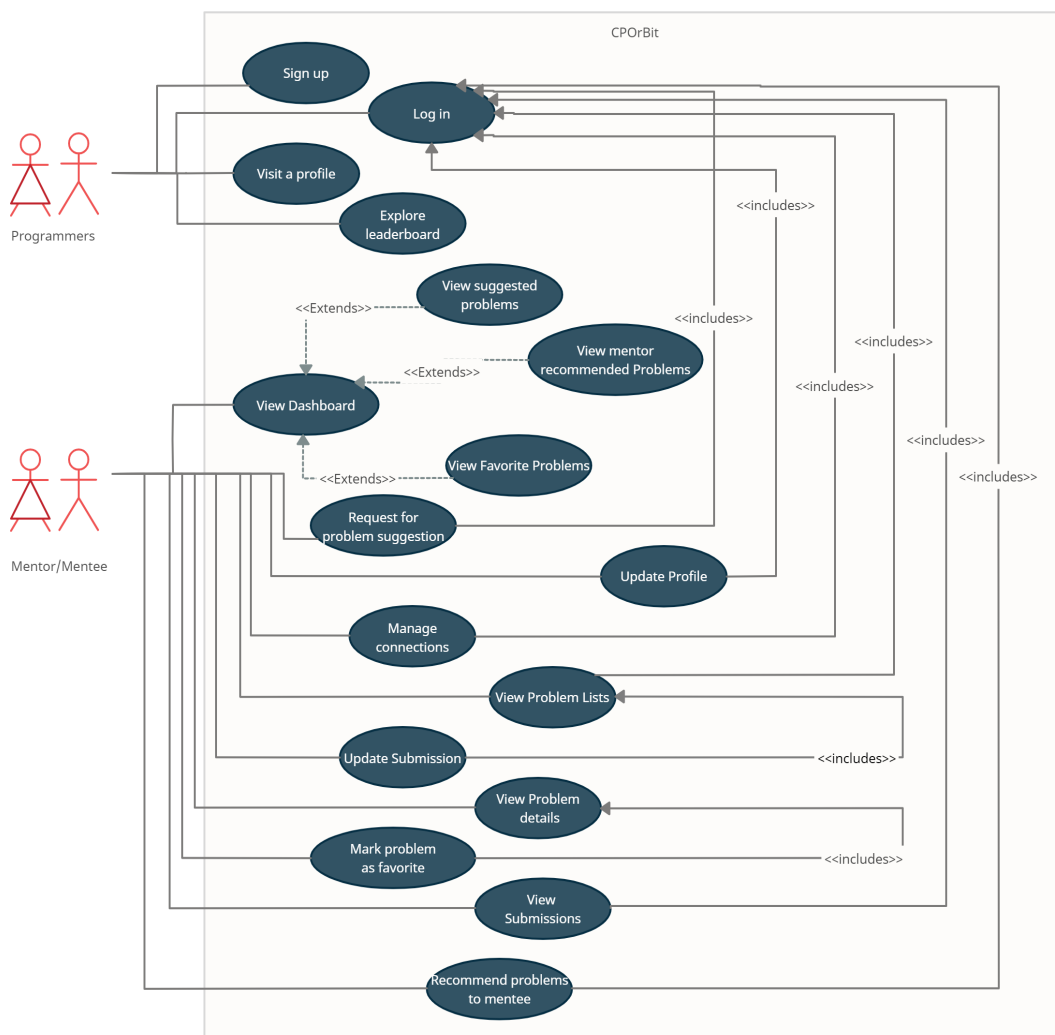


Figure 3.2.1: Use case diagram of CPOrBit

In figure 3.2.1, In the dashboard for a logged-in user the user can access three types of problem lists- problem list that the system suggested, problem list that mentor created, and a list of problems that users marked as favorite.

Table 3.2.1: Use Case Description for Sign Up

Use case Name	Sign up
Actor	Programmers
Precondition	N/A
Internal Path	<ol style="list-style-type: none"> 1. Enter First Name 2. Enter Last Name 3. Enter Email 4. Enter Username 5. Enter Password 6. Enter Confirm Password 7. Click “Sign Up” Button
Exceptional Path	<ol style="list-style-type: none"> 1.1. First Name can’t be blank. 2.1. Last Name can't be blank. 3.1. Must be a valid email. 4.1. Username can’t be empty. 4.2. Must be a unique username. 5.1. Password can’t be blank 6.1. Confirm password must match the password.

Table 3.2.2: Use Case Description for Log In

Use case Name	Log In
Actor	Programmers
Precondition	N/A
Internal Path	<ol style="list-style-type: none"> 1. Enter Username 2. Enter Password 3. Click “Login” Button
Exceptional Path	<ol style="list-style-type: none"> 1.1. Username must be valid. 1.2. Account must be activated. 2.1. Password must match.

Table 3.2.3: Use Case Description for Update Profile

Use case Name	Update Profile
Actor	Mentor/Mentee
Precondition	Log In
Internal Path	1. Enter CF Handle 2. Enter University 3. Upload Profile Picture
Exceptional Path	1.1. Can not be blank. 1.2. Must be valid. 3.1. Must be an image file.

Table 3.2.4: Use Case Description for Request for problem suggestion

Use case Name	Request for problem suggestion
Actor	Mentor/Mentee
Precondition	Log In
Internal Path	1. Click the “Regenerate Suggestion” button.
Exceptional Path	1.1. CF handle must be valid.

Table 3.2.5: Use Case Description for Recommend Problem

Use case Name	Recommend Problem
Actor	Mentor
Precondition	Log In
Internal Path	1. Select Problem 2. Select Mentee 3. Click the “Recommend” button.
Exceptional Path	2.1. Not a mentee

3.3 UML Class Diagram

In the diagram, the main classes are User and Problem. Other classes are Scoreboard, Notification, Profile, Favorite, Tag, Recommend, Suggestion, Submissions. In figure 3.3.1, can authenticate and then using it can access other classes.

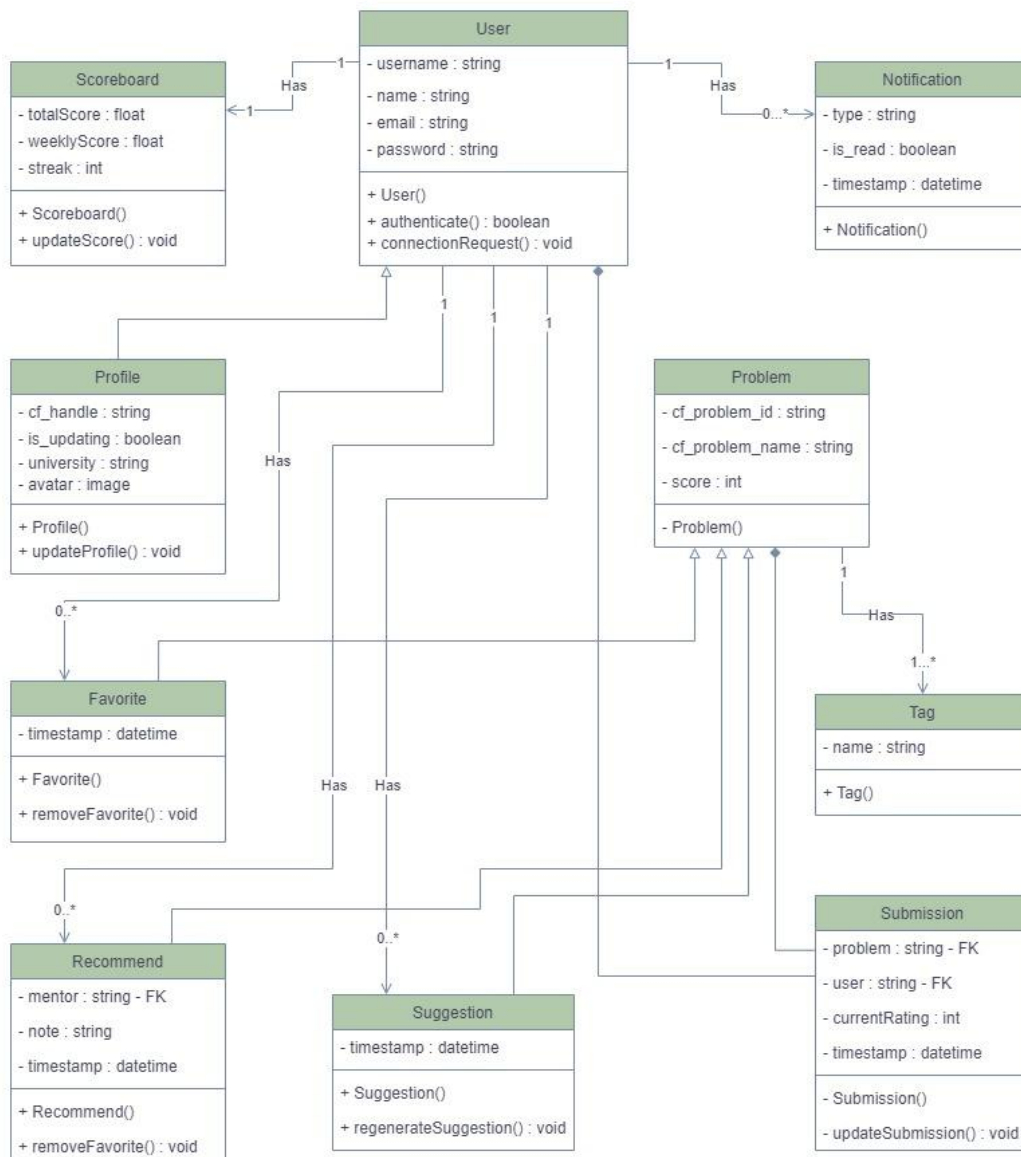


Figure 3.3.1: UML Class Diagram for CPorBit.

Here the minus sign (-) indicates the data and methods are private such as username, password, submission etc. and the plus sign (+) indicates the data and methods are public type such as Favorite, Tag, regenerateSuggestion etc.

3.4 Requirement Collection and Analysis

For this project, the users are competitive programmers who want to improve their performance on a specific platform. To improve their performance we need to first analyze their so far performance data and also analyze the data from other users who are improving. We have shown the Business process model and notation, Use case diagram to show the plans to implement our project. And also in the ER diagram given below, we will get an idea about how we are going to store the data to make it of use. For this project to run Codeforces API[3] must be working. To get more accurate suggestions a user needs more submission. The more we have data points the more accurately we can suggest problems. If there is no submission of a problem, that problem will not be added to our system. And if there is no submission, there will be no statistics to depend on.

3.5 Logical Data Model

The system database has many users. In figure 3.5.1, entities are user Profile, User, Scoreboard, Notification, Tag, Problem, Mentee. Some relational entities are ACsubmissions, Favorite, Recommendation, Suggestion. The Profile has one to one relationship with the User. So mentors/mentees can log in to their profile using email and password. A profile has many attributes like CF handle, university, profile picture etc. Our database is mostly dependent on User entities. A user can solve multiple problems and a problem can have multiple user's submission. A user can have 1 scoreboard. A user can have multiple notifications. A user can have 1 mentor and multiple mentees. A user can have multiple favorite problems, multiple recommended problems and multiple suggested problems. A user can solve multiple problems from multiple tags. A user can recommend multiple problems to another user if he/she is a mentor of that user.

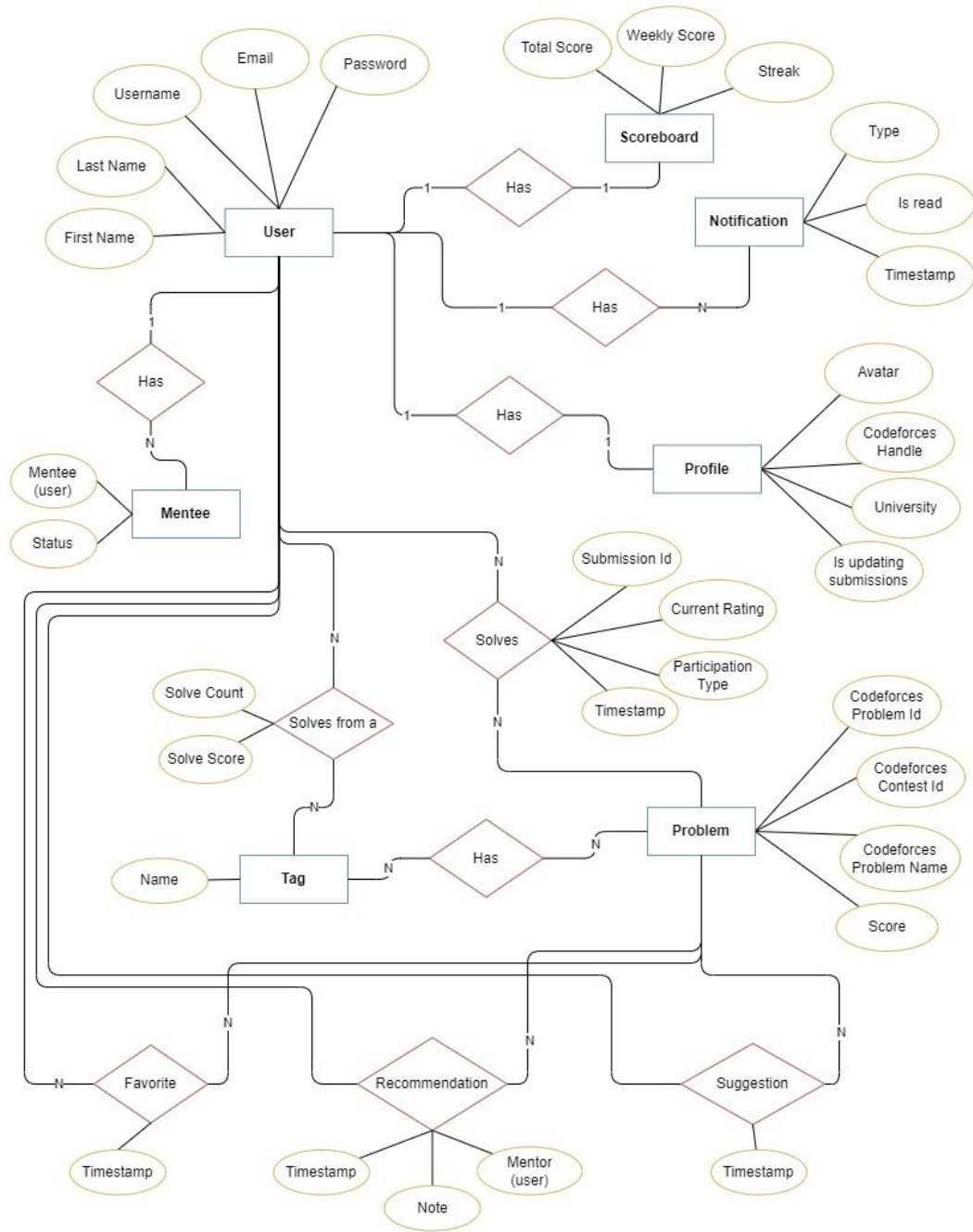


Figure 3.5.1: ER Diagram of CPorBit

3.6 Design Requirements

- We have designed a model to create an account by registering it into the system. We have used Gmail authentication for that.
- We have created a database to store the data of a user so that we can generate a user-exclusive problem list based on their solves.
- We have established a necessary algorithm to suggest problems that are efficient to improve his/her performance.
- We have created a leader board with required data taken from the database to give it a taste of gamification.

We used CSS and Tailwindcss to create the UI. We used drawio[13], an online drawing tool for drawing business process modeling, and UML diagram, creately[12] for drawing the Use case diagram, ER diagram which was functional for planning our progress.

CHAPTER 04

DESIGN SPECIFICATION

4.1 Front End Design

We created an aesthetically pleasing and fashionable interface for our project through UI development. We built a front-end design so that it becomes easier to manage for the user. We have used React.js[15], CSS, TailwindCSS for the front end. We used Javascript in our frontend to interact with the backend server and get the data through API

Landing Page

Figure 4.1.1 and Figure 4.1.2 represent the landing page of our website. This page contains a short introduction and overview about our project. This page has a get started button which will take any logged in user to his/her dashboard. In the navigation bar it has the option to go to the Standing page, Login page, Register page. Landing page can be accessed by anyone even if he/she is not a logged in user of CPorBit.

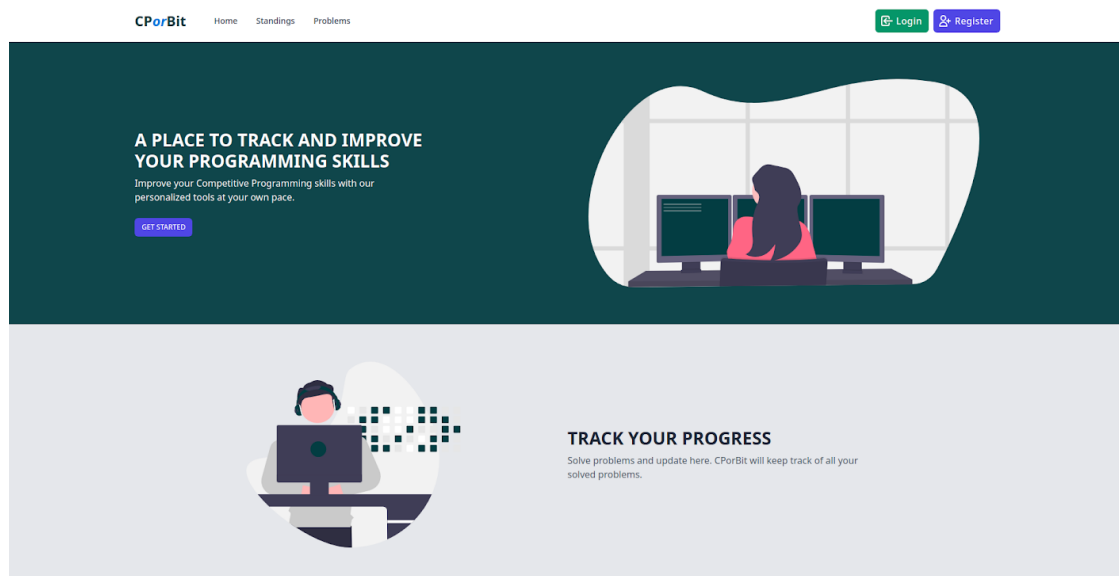


Figure 4.1.1: Screenshot of Landing page for CPorBit

If a user scrolls down he/she can see more details of our website. Here we have added some images and icons so that the purpose of our project can be easily understandable for any type of user.

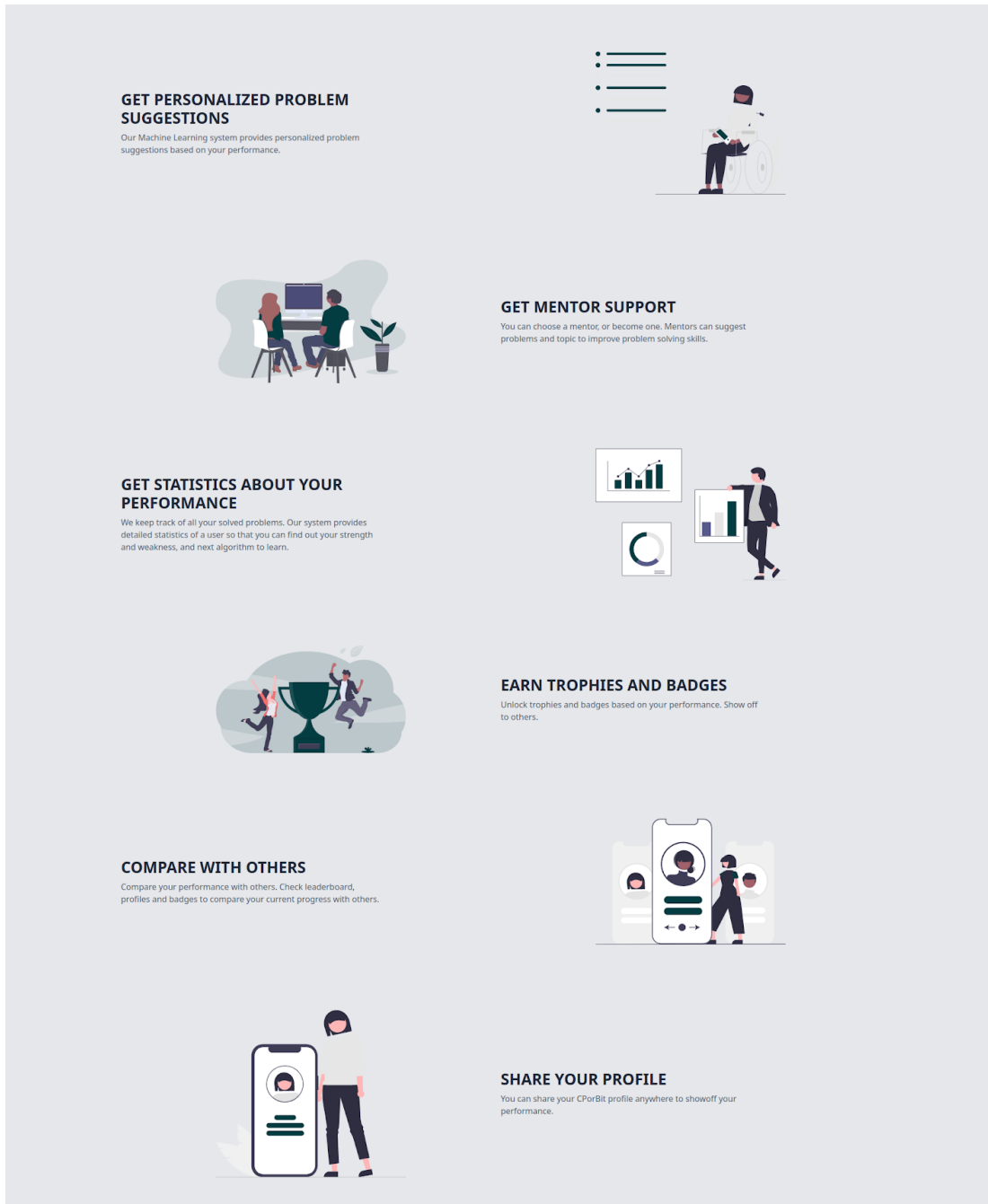


Figure 4.1.2: Screenshot of Landing Page down scrolled

Registration Page

Figure 4.1.3 represents the Registration Page of our website. In this page the user will provide his/her name, username, email address, password and apply for registration. An user activation link will be sent to the provided email. An alert message will be shown to the user after clicking the register button. Below the “Register” button there is a link to the Login Page named “Already have an account? Log in”. Registration Page can be accessed by anyone except logged in users of CPorBit.

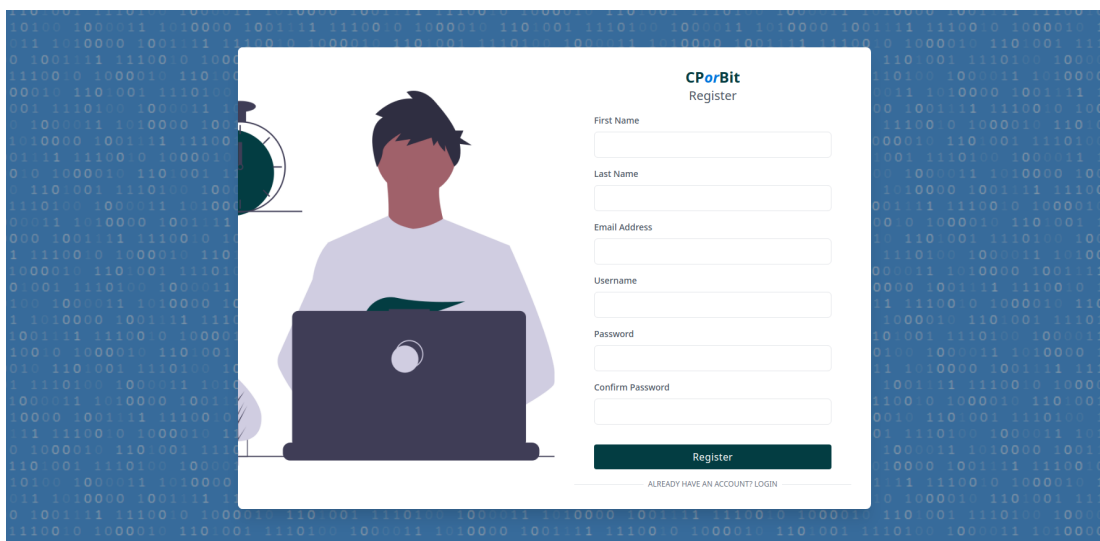
The image shows a registration form for CPorBit. The form is white and centered on a blue background with a binary code pattern. On the left side of the form, there is an illustration of a person with dark hair, wearing a light blue shirt, sitting at a desk with a laptop. The form itself has the following fields: 'First Name', 'Last Name', 'Email Address', 'Username', 'Password', and 'Confirm Password'. Each field is represented by a white rectangular input box. Below the 'Confirm Password' field is a green button with the text 'Register' in white. Underneath the button, there is a link that says 'ALREADY HAVE AN ACCOUNT? LOGIN'.

Figure 4.1.3: Screenshot of Registration Page

Login Page

Figure 4.1.4 represents the Login Page of our website. User will provide his/her username and password and press the “Login” button to login. If the user or password is incorrect then the user cannot login and an alert message will be shown that “username or password is incorrect”. Below the Login button there has a link to Register Page text with “New to CPorBit? Register”. User will get an authentication token, using that user can access all the features available for that user. Login Page can be accessed by anyone except logged in users of CPorBit.

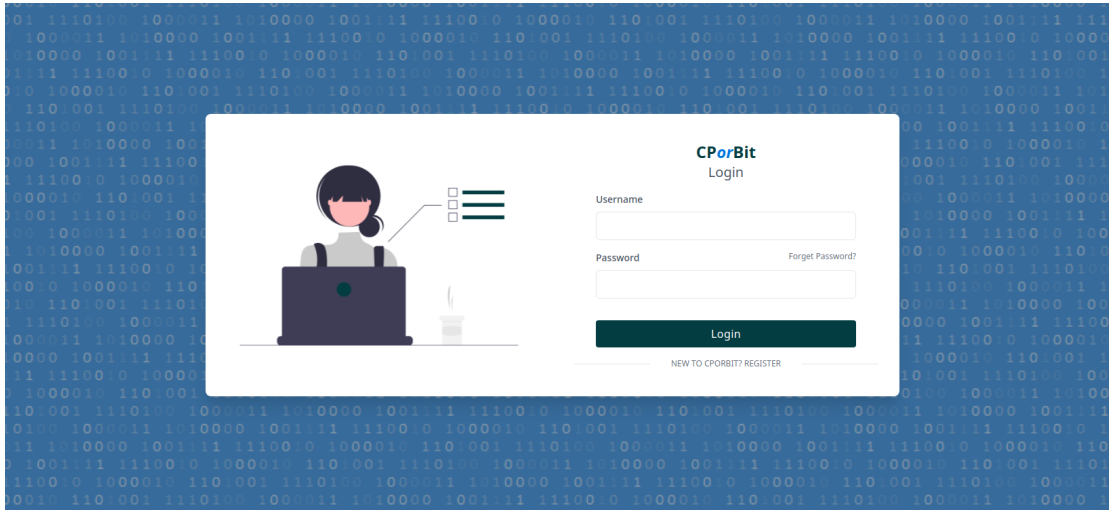


Figure 4.1.4: Screenshot of Login Page for CPorBit

Standings Page

Figure 4.1.5 represents the standing page of our website. The rank of a user is based on the score of that user in descending order. There is a search bar in which users can search any user. By clicking the username of any user, the user can view that user's profile. This page has pagination to move from one page to another. The standing page is visible for anyone even if he/she is not a logged in user of our website.

CPorBit Home Standings Problems [Login](#) [Register](#)

Standings

Search

#	NAME	USERNAME	CF HANDLE	UNIVERSITY	SCORE	STREAK
1	Shah Imran	smhimran	lone_wolf_	Daffodil International University	1229800	5
2	Asif Hasan	asifhasan	asifhasan4566	Daffodil International University	68400	0
3	Shourav Ahmed	ah_med	ah_med	Daffodil International University	7	1
4	Erfanul Islam	erfan	erfanu007	Daffodil International University	2	0
5	Test User	testuser	tanzi5	University of Dhaka	0	0
6	Tanima Hossain	tanima	el.duivel	Daffodil International University	0	0
7	Test User6	user6	sgtlaugh	Some University	0	0
8	Nazmus Sakib	ghost	gh00st	Daffodil International University	0	0
9	Test User5	user5	strange_r	Daffodil International University	0	0

← 1 →

Figure 4.1.5: Screenshot of Standing Page for CPorBit

Profile Page

Figure 4.1.6 represents the profile page which is visible to any other user. The card from left of this page will show user details, scores and streak. If the viewer is logged in and currently doesn't have any connection to that user then there will be two buttons for sending requests to be mentor and mentee. The Submissions button redirects to the user's submission page if the requested user is logged in. Right panel of this page will show a donut chart of users submission statistics. Different colors represent the solve count from different tags of problems. Then Strengths shows the top 5 tags where the user is good at. Weaknesses show the least 5 tags that the user is not very good at. Then the Hardest Solved Problem shows the hardest problem that the user solved.

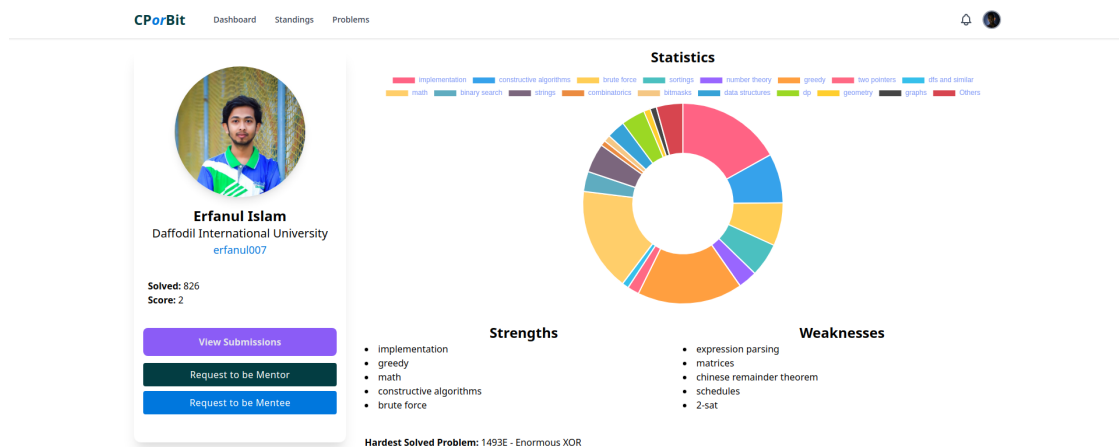


Figure 4.1.6: Screenshot of profile Page for CPorBit

Figure 4.1.7 represents the user's own profile view. From the navigation bar the user can select the view profile option to access this page. Slight difference is that users can click Update Submissions and our website will update all the problems in submission that the user solved from Codeforces. This page also has a View Submissions button to redirect to the submission page. But no button for connection request.

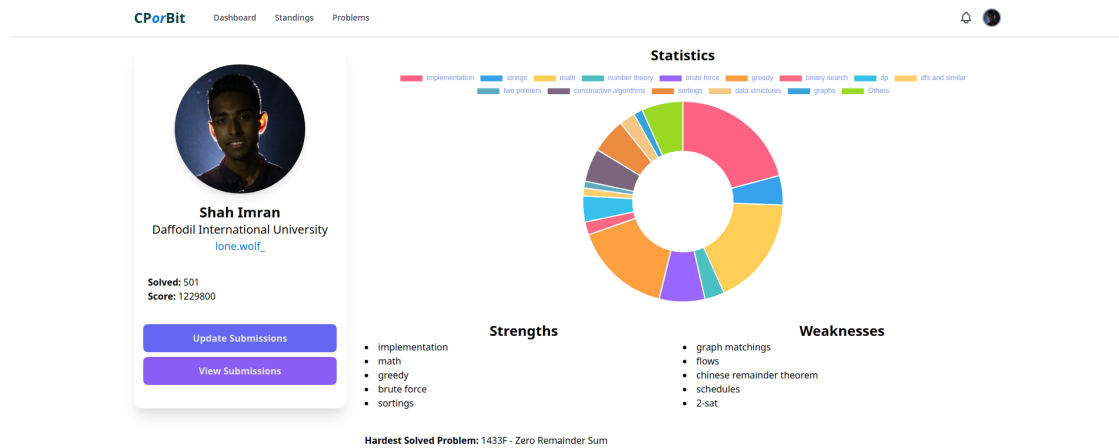


Figure 4.1.7: Screenshot of own profile page for CPorBit

Notification Page

Figure 4.1.8 represents the notification page. There is a notification button in the navigation that shows a pop up of the last 5 notifications. If user clicks show all notifications it redirects to this Notification page where you can see all the notifications and mark all as read or delete all notifications.

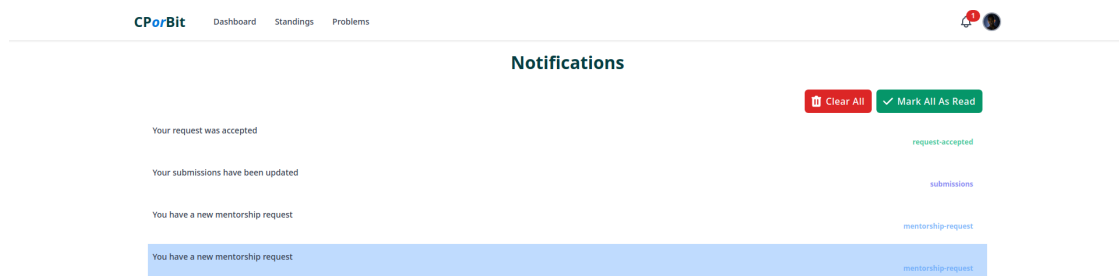
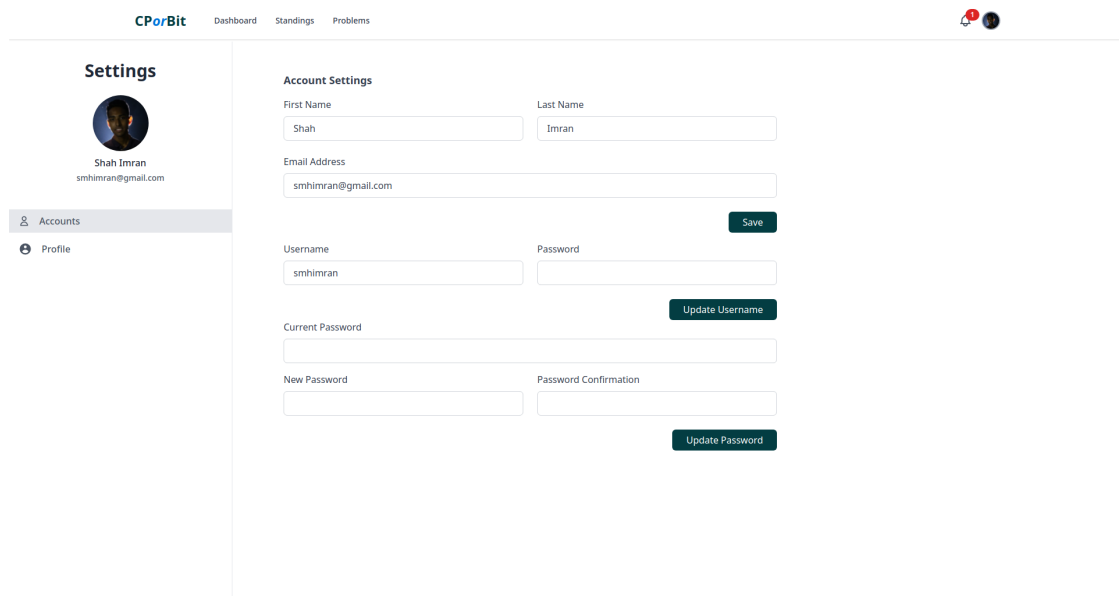


Figure 4.1.8: Screenshot of Notification Page for CPorBit

Account Settings Page

Figure 4.1.9 represents the Account Setting Page. If logged in users can access this page from the pop up of the navigation bar's profile picture by choosing settings option. Users can update personal details like Name, email, username, passwords.

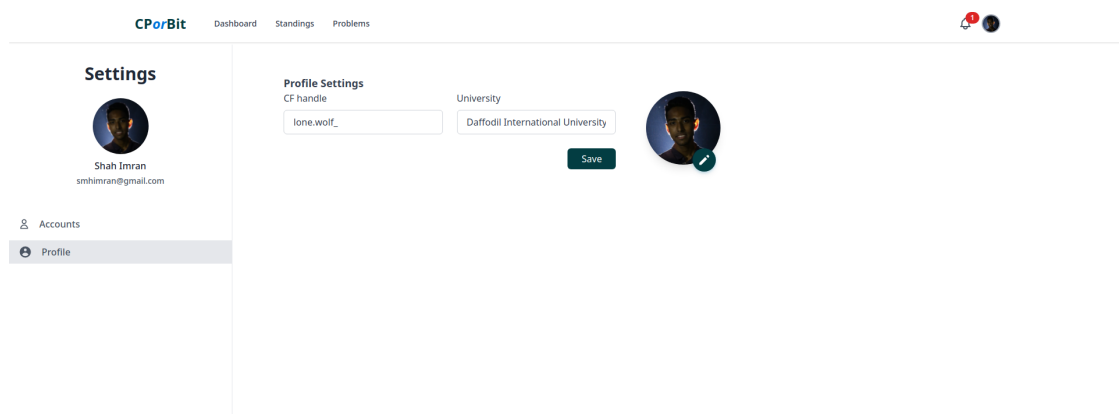


The screenshot shows the 'Account Settings' page for a user named Shah Imran. The page is divided into a sidebar and a main content area. The sidebar contains a profile picture and the name 'Shah Imran' with the email 'smhimran@gmail.com'. Below this are two menu items: 'Accounts' and 'Profile'. The main content area is titled 'Account Settings' and contains several form fields: 'First Name' (Shah), 'Last Name' (Imran), 'Email Address' (smhimran@gmail.com), 'Username' (smhimran), 'Password', 'Current Password', 'New Password', and 'Password Confirmation'. There are three buttons: 'Save' (next to Email Address), 'Update Username' (next to Password), and 'Update Password' (next to Password Confirmation).

Figure 4.1.9: Screenshot of Account Settings Page for CPorBit

Profile Settings Page

Figure 4.1.10 represents the Profile Settings page from settings. In this page users can update their CF handle, university and profile picture.



The screenshot shows the 'Profile Settings' page for the same user. The sidebar is identical to the previous screenshot. The main content area is titled 'Profile Settings' and contains three form fields: 'CF handle' (lone.wolf_), 'University' (Daffodil International University), and a profile picture. There is a 'Save' button below the form fields.

Figure 4.1.10: Screenshot of Profile Settings Page for CPorBit

Current Connections Page

Figure 4.1.11 represents the Current Connection page. Users can access this page by selecting Connections from the navigation bar. This page will show the current mentor, current mentees and give the option to close any connection. This page will also show all the past mentors and mentees list.

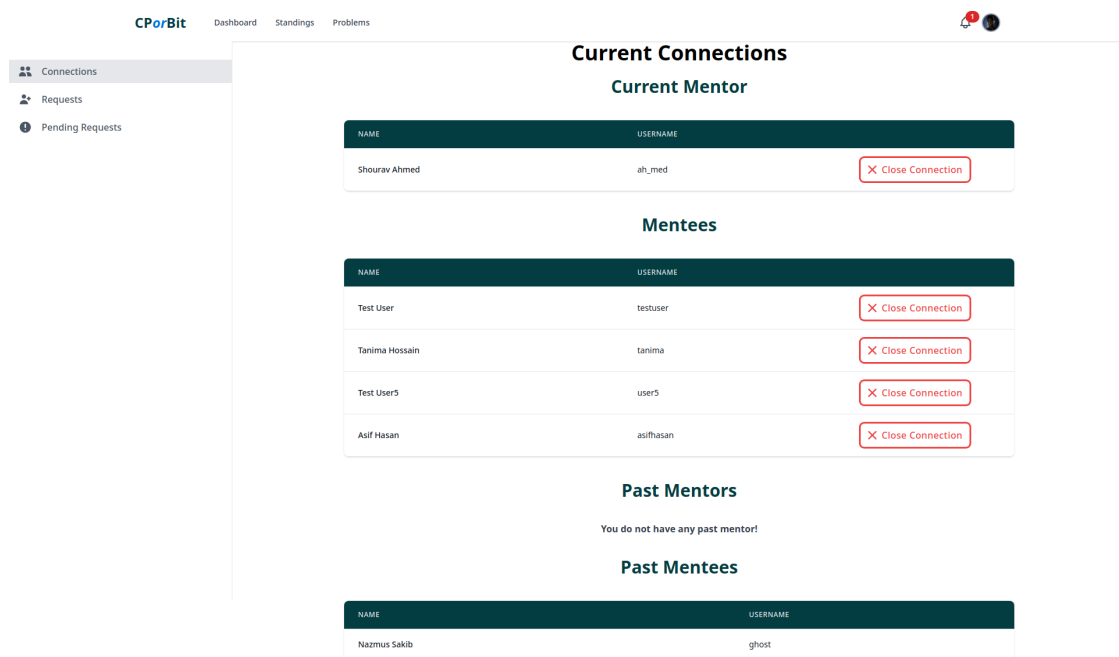


Figure 4.1.11: Screenshot of Current Connections Page for CPorBit

Pending Requests Page

Figure 4.1.12 represents the Pending request page. Users can access this page by selecting Connections from the navigation bar and then Pending Requests from left navigation. Users can see all the requests he/she sent to others and cancel those requests.

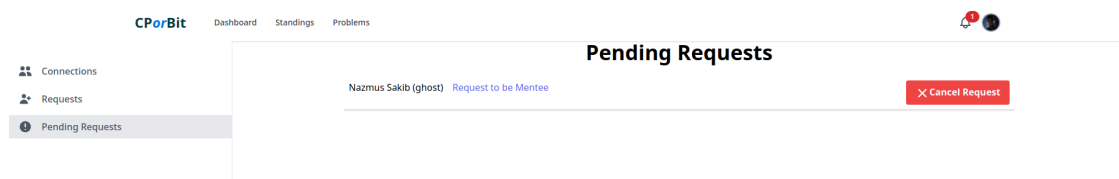


Figure 4.1.12: Screenshot of Current Pending Requests Page for CPorBit

Requests Page

Figure 4.1.13 represents the Request page. Users can access this page by selecting Connections from the navigation bar and then Requests from left navigation. This page will show all the requests the user received to be a mentor or mentee. Users can accept or reject requests from here. If rejected, the request will be deleted. If accepted, the connection will update for that user.

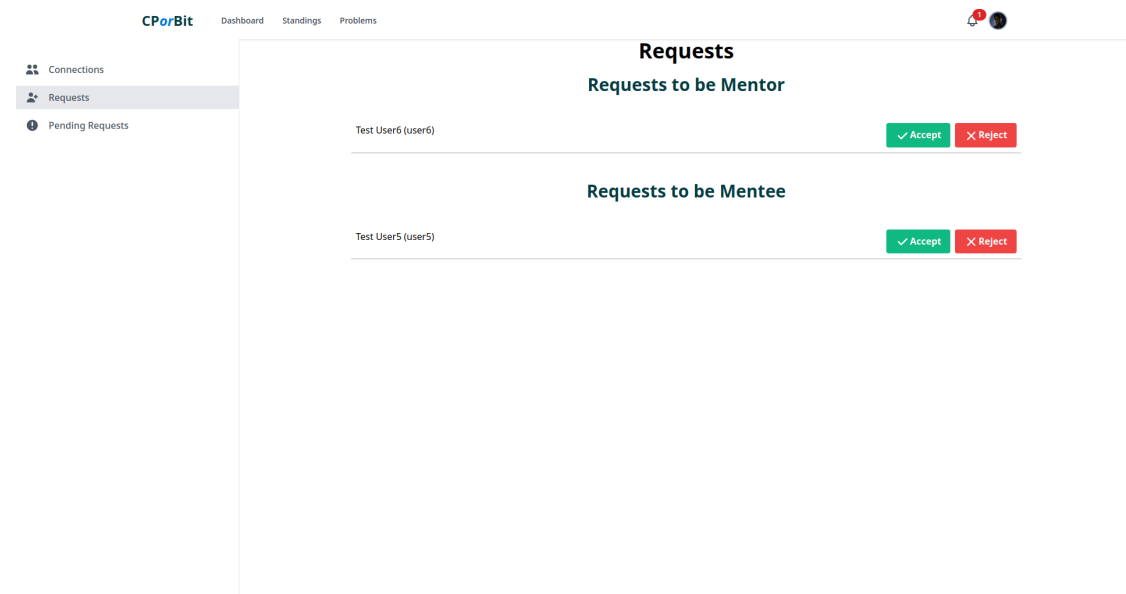


Figure 4.1.13: Screenshot of Current Requests Page for CPorBit

All Problem Page

Figure 4.1.14 represents the All Problem page. Here a user can view all the problems, their total solved in this system, score, tags and if the user solved that problem or not. Problem row will have a green shade if that problem is solved by the requested user. For this user must be logged in. Users can view the problem page through clicking any problem from this list. There is a search bar at the top. Users can search any problem by entering CF problem id and click search or enter..

PROBLEM	SCORE	PEOPLE SOLVED	SOLVED
1003A - Polycarp's Pockets Implementation	800	3	✓
1003B - Binary String Constructing constructive algorithms	1300	3	✓
1003C - Intense Heat brute force math Implementation	1300	2	
1003D - Coins and Queries greedy	1600	1	
1004A - Sonya and Hotels Implementation	900	2	
1004B - Sonya and Exhibition math implementation greedy constructive algorithms	1300	2	
1004C - Sonya and Robots Implementation constructive algorithms	1400	2	
1005A - Tanya and Stairways Implementation	800	3	✓

Figure 4.1.14: Screenshot of All Problems Page for CPorBit

Problem Page

Figure 4.1.15 represents the problem page. Here users can see details of any problem. That includes CF problem id, name, tags, scores, solve count from our system. It will show if it is a suggested problem and recommended problem.

1003B - Binary String Constructing

constructive algorithms

★ System Suggested Recommended by Mentor

Score: 1300

👤 Solved By: 3

🟢 Solved

Recommend this problem to mentees

- Test User (testuser) [Recommend](#)
- Tanima Hossain (tanima) [Recommend](#)
- Test User5 (user5) [Recommend](#)
- Asif Hasan (asifhasan) [Recommend](#)

Figure 4.1.15: Screenshot of Problem Page 1

Figure 4.1.16 represents the same problem page but for unsolved, not recommended, not suggested and favorite. Users can mark/unmark a problem as their favorite by clicking the star icon.

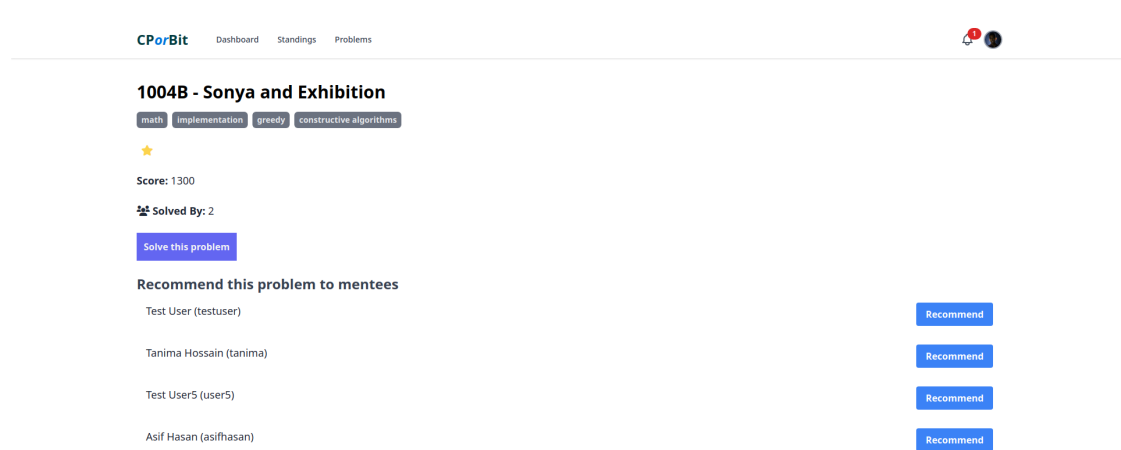


Figure 4.1.16: Screenshot of Problem Page 2

Dashboard

Figure 4.1.17 represents the dashboard of our website. This is the default landing page for logged in users. This page will show all the system suggested problems on the left side. Users can click on the problem name and access the problem page. The row will have shade of green if the problem is solved. Similarly the right side will show the recommended problems and favorite problems. Here, view all recommendations links will lead the user to all the recommended problems by mentor. And view all favorites links will lead the user to all the problems that the user marked favorite. Both have the same functionalities as suggested problems. Gen New Suggestion button will update all the system suggested problems for the user.

The screenshot shows the CPorBit dashboard with two main sections: "System Suggested Problems" and "Recommended Problems".

System Suggested Problems:

- 1119A - Ilya and a Colorful Walk (Implementation, greedy)
- 1029B - Creating the Contest (math, greedy, dp)
- 1131B - Draw! (Implementation, greedy)
- 1100A - Roman and Browser (Implementation)
- 1388B - Captain Flint and a Long Voyage (math, greedy)
- 1087B - Div Times Mod
- 1058B - Vasya and Cornfield (geometry)
- 1037C - Equalize (strings, greedy, dp)
- 1107B - Digital root (math, number theory)

Recommended Problems:

- 1037B - Reach Median (greedy)
- 1003B - Binary String Constructing (constructive algorithms)
- 1016A - Death Note (math, Implementation, greedy)
- 1003D - Coins and Queries (greedy)
- 1033C - Permutation Game (brute force, dp, games)

Favorites:

- 1008A - Romaji (Implementation, strings)
- 1004B - Sonya and Exhibition (math, Implementation, greedy, constructive algorithms)

Figure 4.1.17: Screenshot of Dashboard

Submission Page

Figure 4.1.18 represents the submission page of an user. Any logged in user can view all submissions of anyone by clicking the View Submission button from the profile page. This page contains details of all the accepted submissions of a user.

The screenshot shows the "Submissions of smhimran" page. It features a table with the following columns: SUBMISSION ID, PROBLEM, TIME, and PARTICIPATION TYPE.

SUBMISSION ID	PROBLEM	TIME	PARTICIPATION TYPE
141304224	1008A - Romaji	2022-01-01T10:23:13Z	PRACTICE
133764628	1604D - Moderate Modular Mode	2021-10-31T12:32:10Z	PRACTICE
133668338	1604C - Di-visible Confusion	2021-10-30T15:58:11Z	CONTESTANT
133654032	1604B - XOR Specia-LIS-t	2021-10-30T15:27:18Z	CONTESTANT
133633946	1604A - Era	2021-10-30T14:54:59Z	CONTESTANT
131945653	1593D1 - All are Same	2021-10-14T13:59:08Z	PRACTICE
131810599	1593C - Save More Mice	2021-10-13T15:42:24Z	CONTESTANT
131793117	1593B - Make it Divisible by 25	2021-10-13T15:19:54Z	CONTESTANT
131765250	1593A - Elections	2021-10-13T14:50:51Z	CONTESTANT
124398658	1554B - Cobb	2021-07-31T07:15:05Z	PRACTICE
124397554	1554A - Cherry	2021-07-31T07:01:42Z	PRACTICE
123456200	1551A - Polycarp and Coins	2021-07-23T14:45:23Z	CONTESTANT
123402139	1553B - Reverse String	2021-07-23T07:01:55Z	PRACTICE
123294848	1553A - Digits Sum	2021-07-22T14:44:15Z	CONTESTANT

Figure 4.1.18: Screenshot of Submissions

DarkTheme

Figure 4.1.19 represents the view of the dark theme. Users can enable the dark theme or the light theme from the navigation bar. All the pages of this website adopt the theme's view when it is selected.

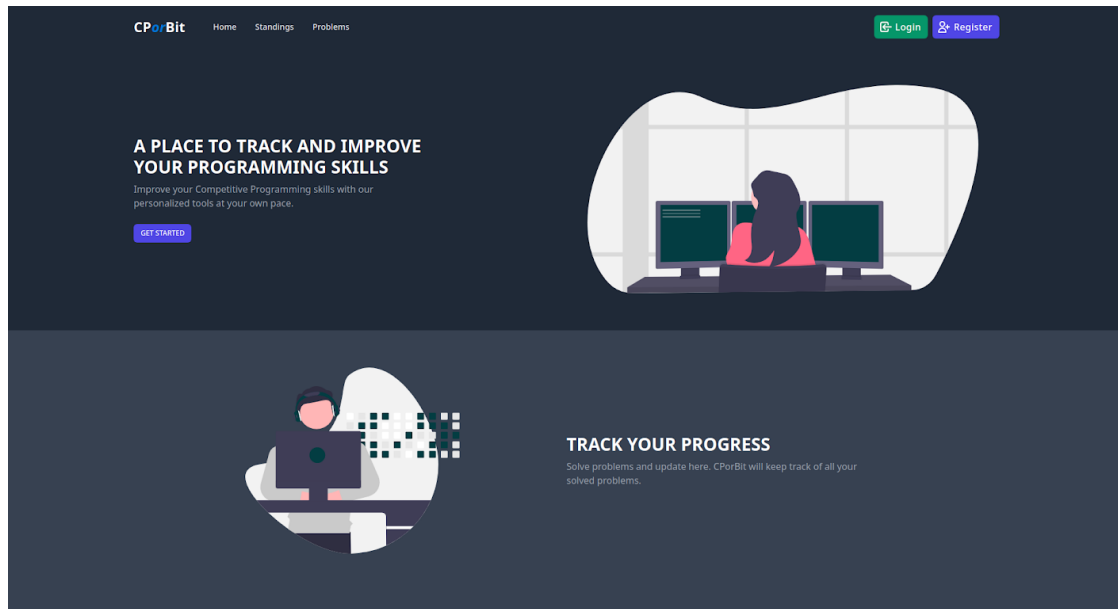


Figure 4.1.19: Screenshot of Dark Theme for CPorBit

4.2 Back-End Design

In our project, front-end and back-end are totally different applications. They interact between themselves through API. CPorBit's back-end is developed with Django-Rest Framework[14]. The back-end is divided into different modules. In Django these are called applications. Our project has user app, problem app, suggestion app and gamification app. For development and testing we used sqlite database. But in production the database is PostgreSQL[21]. MVC was used for fast and clean development.

4.3 Interaction Design and UX

- Design: To make this project modern stack has been used to make it up to date and easily modifiable over the time if needed. The UI is created to be user friendly so that someone can easily enjoy the experience from the very first time.

- Completion: The project very much auto manages itself. So it is going to be up to date all the time.
- Project Deliverables: The project is going to be more and more efficient at its job depending on more data it will get. And in the future it has the scope of adding machine learning to it.
- Resource Allocation: The resource it will need and use is data only. The more data it will get from the user the more accurate data it will provide to the user as well.

4.4 Implementation Requirements

Many things are required to implement our project. Main and foremost thing is Django-Rest Framework[14] which is the backend framework of our project. Python[16] is the language of this framework. For databases we need PostgreSQL[21]. Developers need to have a good understanding of OOP and MVC. Then Node.js needed to run the frontend of the website and get all the data through API calls using Javascript. We used React[15] as our website is a single page application. CSS and Tailwind are required to design the front end. Git[19] is very useful to have for managing the project.

Software Requirements

We need some software for coding, implementing, debugging, testing and controlling the application.

- Visual Studio Code - for smart coding and debugging[18]
- Postman - for testing API[17].
- Chrome / Any Web Browser - for interacting with the frontend.

CHAPTER 05

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

We have made every effort to ensure the security and management of the application in the database. The database must be able to handle a large number of data, with a lot of transactions. Since there will be a lot of updates of problem details, solve status, and statistics, the database has to be good and reliable for storing data. We have chosen PostgreSQL[21], which is a free, open-source, relational database management system emphasizing extensibility and SQL compliance. As our backend is made with Python[16], PostgreSQL[21] is a good choice since it works very well with Python[16]. Our database server is hosted on the same server the application is hosted with, so there is a very low latency between the two servers.

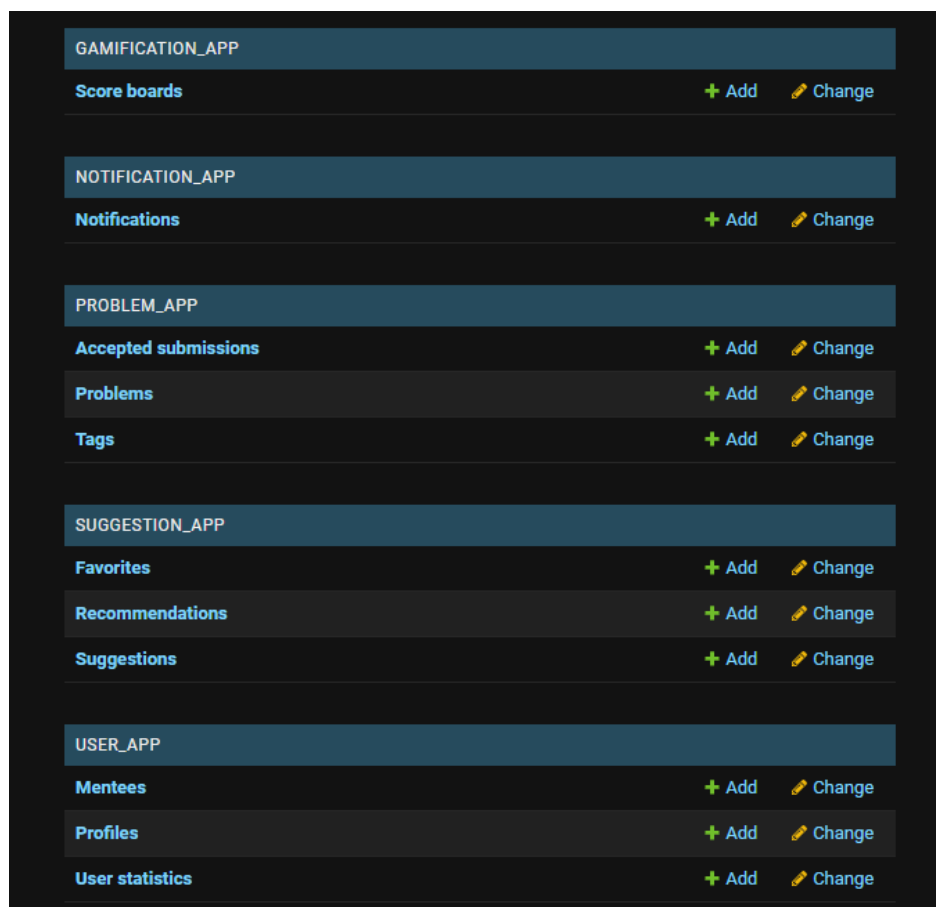


Figure 5.1.1: Screenshot of Database Tables of CPorBit

5.2 Implementation of Back-end

Our application will be used by a lot of users. So, it has to be fast, secure, reliable, and scalable. We also want our application to be able to provide services for different applications such as a mobile application, or a different web application. So we decided to create the backend using the REST API. We have used Django, which is an awesome framework for creating web applications. It provides great Authentication and Authorization methods, scalability, and is very good at handling a lot of users. To create the REST APIs, we have used Django-REST Framework[14]. We have used token-based authentication in our application to make it RESTful. Finally, the suggestion-generating part was created using Python[16].

5.3 Implementation of Front-end Design

As the backend is RESTful, it is very convenient to use a Single Page Application (SPA) as the front-end. So we have chosen React[15] to build the application. We have used React's Context API for data management. To design and create the UI, we used CSS and Tailwindcss, which is a UI utility library to make beautiful UI with fewer efforts. We have used Axios, a library to make HTTP requests, communicate with the backend, and fetch data. We have made the application responsive, to be able to use on various devices such as computers, smartphones, and tablets.

5.4 Testing Implementation

Developing the application is not complete unless it has been tested to ensure that it is working properly. That is why we have performed unit testing after each feature is created to make sure that it is ready for use. Also, after adding some features, we have performed integration testing combining some features. Since we are not familiar with any automated testing tools, we have tested the application manually, by ourselves.

Table 5.4.1: Test Cases

No.	Test Case	Test Input	Expected Outcome	Obtained Outcome	Pass/Fail
1	User Registration	Users must fill the sign up form by name, email, username, password and other necessary data.	Send Email Verification	Email Verification Sent	Pass
2	Activate Account	Check the email provided during registration and click the link sent in the mail.	Account Activated.	Account Activated.	Pass
3	User Log In	User must fill the login form using correct username and password	Account logged in	Account logged in	Pass
4	User Log In	User filled wrong information	Username or password is incorrect.	Username or password is incorrect.	Pass
5	User Log In	Filled information of an unactivated account.	Account not activated.	Account not activated.	Pass
6	Update Submission	Clicking update submission button without updating valid CF handle	Didn't update correct data.	Didn't update correct data.	Pass
7	Update Submission	Clicking update submission button after updating valid CF handle.	Got all the submissions and updated the problems and tags.	Got all the submissions and updated the problems and tags.	Pass
8	Sent a connection request	Send a request from someone's profile.	Received request from that user	Received request from that user	Pass

9	Accept a connection request	Click the accept button for a request.	Connection created.	Connection created.	Pass
10	Cancel a connection request	Click the cancel button for a request	Request deleted.	Request deleted.	Pass
11	Remove Connection	Click remove button for a connection	Connection closed.	Connection closed.	Pass
12	Add a favorite	Mark a problem as favorite from the problem details.	Problem added to the favorite list.	Problem added to the favorite list.	Pass
13	Remove a favorite	Unmark a problem as favorite from the problem details.	Problem removed from the favorite list.	Problem removed from the favorite list.	Pass
14	Generate Suggestion	Click the generate suggestion button	Update all suggestions correctly for this user.	Update all suggestions correctly for this user.	Pass
15	Recommend Problem	Select mentee and click the recommend button from problem details.	Problem added to the recommended list of that mentee.	Problem added to the recommended list of that mentee.	Pass
16	Remove recommendation.	Select the problem from the recommendation list of a mentee and click the remove button.	Problem removed from the recommended list of that mentee.	Problem removed from the recommended list of that mentee.	Pass
17	Resolve recommendation	Mark a problem as completed from the recommendation list by the mentor when the problem is solved.	Problem removed from the recommended list.	Problem removed from the recommended list.	Pass

18	Search User	Provide valid user name in the search bar of the standing page.	User found.	User found.	Pass
19	Search problem	Provide Valid CF problem ID in the search bar of the problem page.	Problem found.	Problem found.	Pass
20	Mark notification as read	Click the mark as read button of the notification page.	Marked all notifications as read.	Marked all notifications as read.	Pass

5.5 Test Results and Reports

The test results and reports are satisfactory and we can publish this application for people to use. Though any application is not above bugs and failures, we have to ensure that it is working as expected, with the least issues. Since we have performed manual testing, there might be some issues that were overlooked. But we can rest assured that the application is working fine for regular use and is ready to be published.

CHAPTER 06

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

This project would make a programmer's journey in competitive programming a lot easier and interesting. A user will get to know their strengths and weaknesses, current stats, and compare themselves with other users. They will get problem suggestions to solve, to improve their performance. Those problems will be based on their strengths, weaknesses, and current performance. This is a lot helpful as a competitive programmer, getting a good suggestion on what to solve, on which topic. Moreover, the user will be able to choose other users as mentors or mentees. A mentor can suggest problems to a user outside the system suggestions. In addition, to make the users' problems solving more interesting, and make sure they don't get bored of solving suggested problems, they will be rewarded with points after solving a system-generated problem. These points will be used to form the leaderboard, where a user will be able to see their position, compare themselves with other users. However, it lacks a few things that should be added soon. Such as, there is no way to communicate with other users, even mentors/mentees directly. Also, the system problem generation can be improved using some better machine learning models. We'll be working on adding such features to make the application more reliable and useful.

6.2 Limitations of Our Application

As with all applications, our application also has some limitations. We're working on these issues to make the application more useful. Some of the most significant drawbacks are:

- This is a web application. So, the application can only be accessed from a web browser connected to the internet.

- As there is no fixed way for a programmer to solve a problem, the suggestions may not always be 100% accurate.
- There is no way to communicate with other users directly.
- There are no resources available to learn specific topics. The users will get problem suggestions based on topics, but they'll not be provided any resources to solve and learn the related topics.
- The application depends on only one online judge, Codeforces.

6.3 Scopes for Future Developments

We have visions to overcome our limitations and make the application better and more sustainable. Scopes for future development are below:

- We will build a mobile application so that the users don't have to rely on a web browser. But the web version will still be available.
- We will introduce a messaging system. A user will be able to communicate with other users to direct messages.
- To improve the suggestions, we will use some machine learning algorithms to analyze user statistics and generate suggestions more effectively.
- We will add a resources section, where a user will find some articles, tutorials about the most common topics. This section will be open to all, and a registered user can add some resources about some topics.
- We will add a forum for the users, where they can discuss various topics and problems. This section will be open for everyone to view, but only registered users can add a thread, comment, or reply.
- We will add other online judge support so that the system and users don't have to depend on just one online judge.
- The system will be a full-fledged online judge, with its own problems submitted by users, and self code execution system.

These future plans can be executed if proper support and resources are available

REFERENCES

- [1] Code-drills (for problem recommendation), available at <<<https://recommender.codedrills.io/>>> last accessed on 2nd November, 2021 at 2:23PM.
- [2] Codeforces (for all details of profile), available at <<<https://codeforces.com/>>> last accessed on 4th November, 2021 at 7:51PM.
- [3] Codeforces API (for API documentation), available at <<<https://codeforces.com/apiHelp>>> last accessed on 28th November, 2021 at 3:00PM.
- [4] Codeforces Rating System, available at <<<https://codeforces.com/blog/entry/20762>>> last accessed on 8th September, 2021 at 10:56PM.
- [5] Codeforces Rating System update 01, available at <<<https://codeforces.com/blog/entry/77890>>> last accessed on 1st October, 2021 at 1:35AM.
- [6] Codeforces Rating System update 02, available at <<<https://codeforces.com/blog/entry/79727>>> last accessed on 4th October, 2021 at 3:07PM.
- [7] CodeFlares (down at this moment), available at <<<https://www.codeflares.com/>>> last accessed on 16th July, 2021 at 10:50PM.
- [8] Hackerrank (For Problem Recommendation), available at <<<https://www.hackerrank.com/>>> last accessed on 5th October, 2021 at 5:22AM.
- [9] StopStalk (For Problem Recommendation), available at <<<https://www.stopstalk.com/>>> last accessed on 15th October, 2021 at 9:37PM.
- [10] CodeChef (For difficulty analysis), available at <<<https://www.codechef.com/problems/school>>> last accessed on 16th October, 2021 at 11:34AM.
- [11] Spoj (For Problem Difficulty), available at <<<https://www.spoj.com/problems/classical/>>> last accessed on 22nd October, 2021 at 1:25AM.
- [12] Creately (For drawing diagram), available at <<<https://app.creately.com/>>> last accessed on 25th October, 2021 at 9:55PM.
- [13] Drawio (For drawing diagram), available at <<<https://app.diagrams.net/>>> last accessed on 5th November, 2021 at 10:45PM.
- [14] DRF (For development), available at <<<https://www.django-rest-framework.org/>>> last accessed

on 12th November, 2021 at 6:49PM.

[15] React (For development), available at <<<https://reactjs.org/docs/getting-started.html>>> last accessed on 14th November, 2021 at 12:27AM.

[16] Python (For development), available at <<<https://www.python.org/doc/>>> last accessed on 17th November, 2021 at 11:15AM.

[17] Postman (For development and testing), available at <<<https://www.postman.com/>>> last accessed on 23th November, 2021 at 7:55PM.

[18] Visual Studio Code (For development, testing), available at <<<https://code.visualstudio.com/>>> last accessed on 7th December, 2021 at 9:10PM.

[19] Git (For development), available at <<<https://git-scm.com/>>> last accessed on 24th December, 2021 at 1:26AM.

[20] Stack Overflow (For help), available at <<<https://stackoverflow.com/>>> last accessed on 22nd December, 2021 at 4:52PM.

[21] PostgreSQL (For Database), available at <<<https://www.postgresql.org/>>> last accessed on 30th December, 2021 at 1:49PM.

APPENDICES

Name	Description
CF	CodeForces - is an online contest platform.
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DRF	Django-Rest Framework
BPMN	Business Process Model and Notation
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
API	Application Programming Interface
OOP	Object Oriented Programming
ORM	Object-Relational Mapping
MVC	Model-View-Controller
SPA	Single Page Application
App	Application
CSE	Computer Science and Engineering

Turnitin Originality Report

Processed on: 01-Jan-2022 19:19 +06
 ID: 1736748983
 Word Count: 8496
 Submitted: 1

Similarity Index 17%	Similarity by Source Internet Sources: 16% Publications: 3% Student Papers: 13%
--	---

CPorBit : An Automated Mentor For Competitive Programmers By Md. Erfanul Islam Bhuiyan

2% match (student papers from 20-Jan-2021) Class: Fall 2020 - Fall 2023 Assignment: S1 Paper ID: 1490618914
1% match (Internet from 01-Oct-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5284/171-15-1272%3d20%25.docx?isAllowed=y&sequence=1
1% match (Internet from 17-Sep-2019) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3314/P13236%20%289%25%29.pdf?isAllowed=y&sequence=1
1% match (Internet from 05-Oct-2019) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3319/151-15-242%3d28%25%20CSE.pdf?isAllowed=y&sequence=1
1% match (Internet from 01-Jun-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5376/162-15-8175%20%2816_%29.pdf?sequence=1
1% match (Internet from 07-Apr-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5428/173-15-10312%20%2811%25%29.pdf?isAllowed=y&sequence=1
1% match (student papers from 02-Apr-2019) Submitted to Daffodil International University on 2019-04-02
1% match (student papers from 14-Jan-2021) Class: Fall 2020 - Fall 2023 Assignment: S1 Paper ID: 1487304266
< 1% match (Internet from 23-Nov-2019) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2760/P12235%20%2814_%29.pdf?sequence=1
< 1% match (Internet from 02-Apr-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5208/162-15-7927%20%2818_%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 07-Apr-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5163/P14912%20%2817_%29CSE.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 07-Apr-2021) http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5506/163-15-8485%20%2827_%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 07-Apr-2021)