PROJECT NAME – CODE CASE

DESIGNED BY – HARSH KUMAR

ROLL – IMT2021016

TECHNOLOGY USED – MONGODB, REACTJS, NODEJS, EXPRESSJS

DESCRIPTION – Codecase is a web application designed to improve your competitive programming skills. The app provide 3 options.

1.) Random practice
2.) Create mashup contest
3.) Practice weak concepts

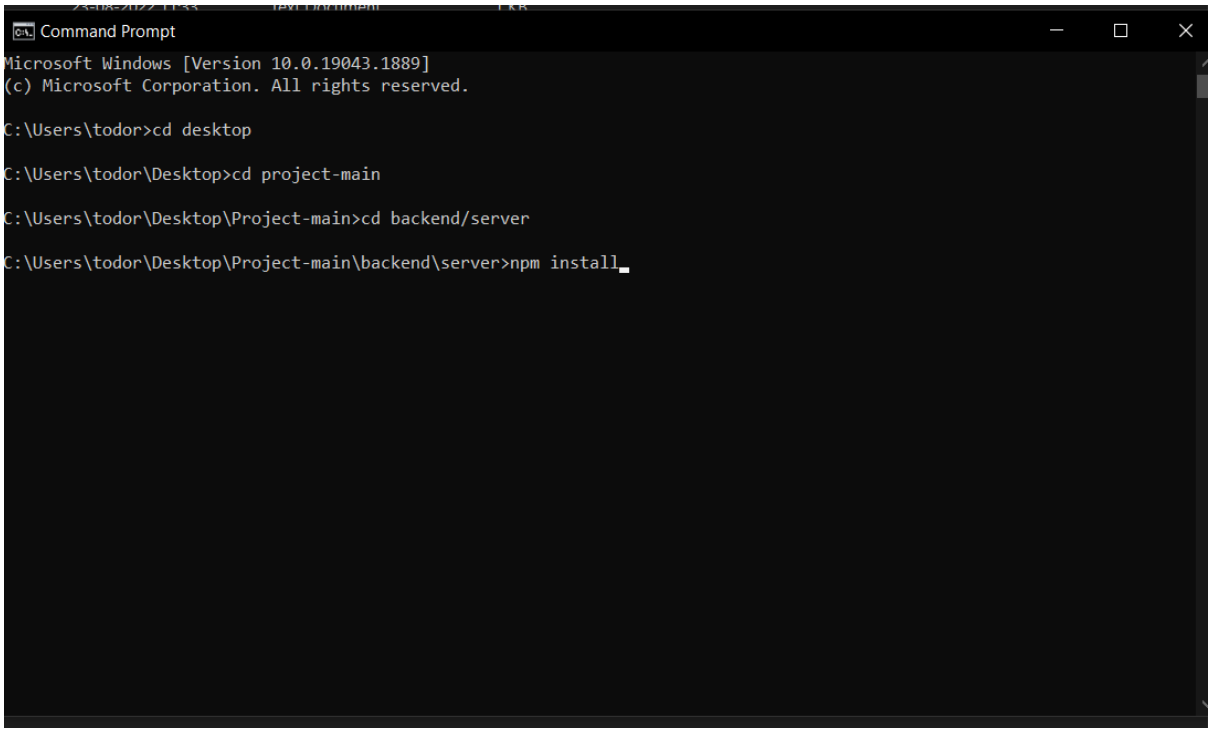
IN RANDOM PRACTICE, questions are selected based on your rating to improve your problem solving skill. If your rating is “x” then questions selected will be random in range [x, x+400]. This will ensure that the problem you practice are challenging and at the same time suited for your skill.

IN CREATE MASHUP CONTEST, contest is created based on the rating of participatnts, so that questions are solvable as well as challenging for everyone.

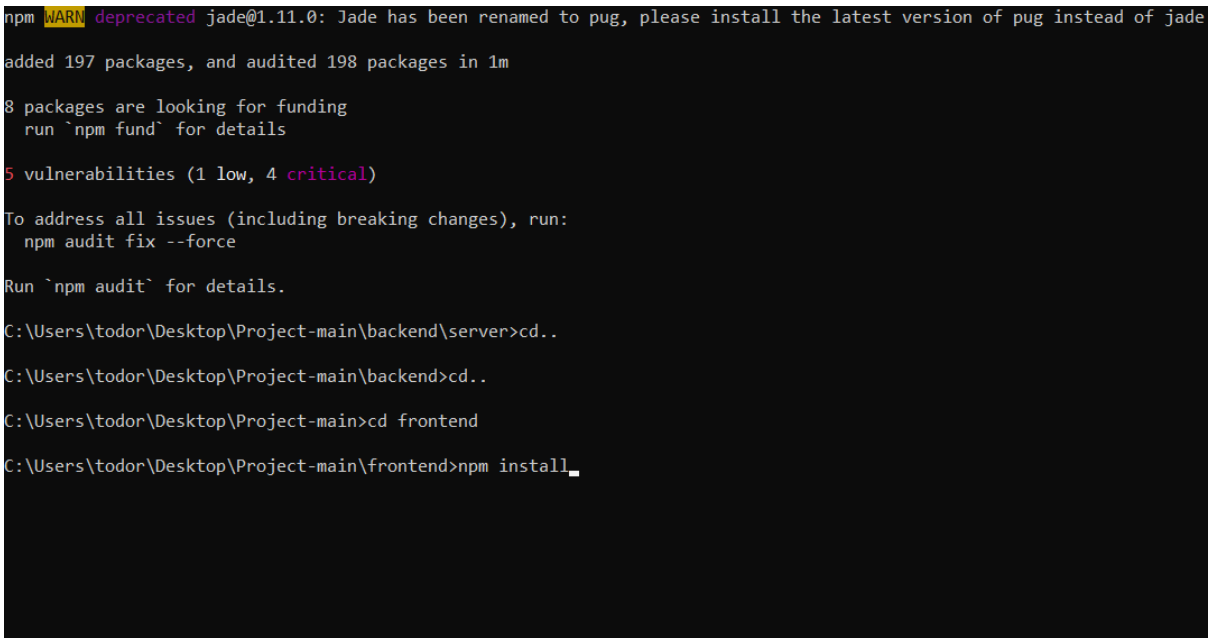
IN PRACTICE WEAK CONCEPTS questions are provided based on you unsolved problem. After looking at your unsolved problem list, weak topics are decided and questions are selected based on most weak topic to least weak topic.

Starting the application.

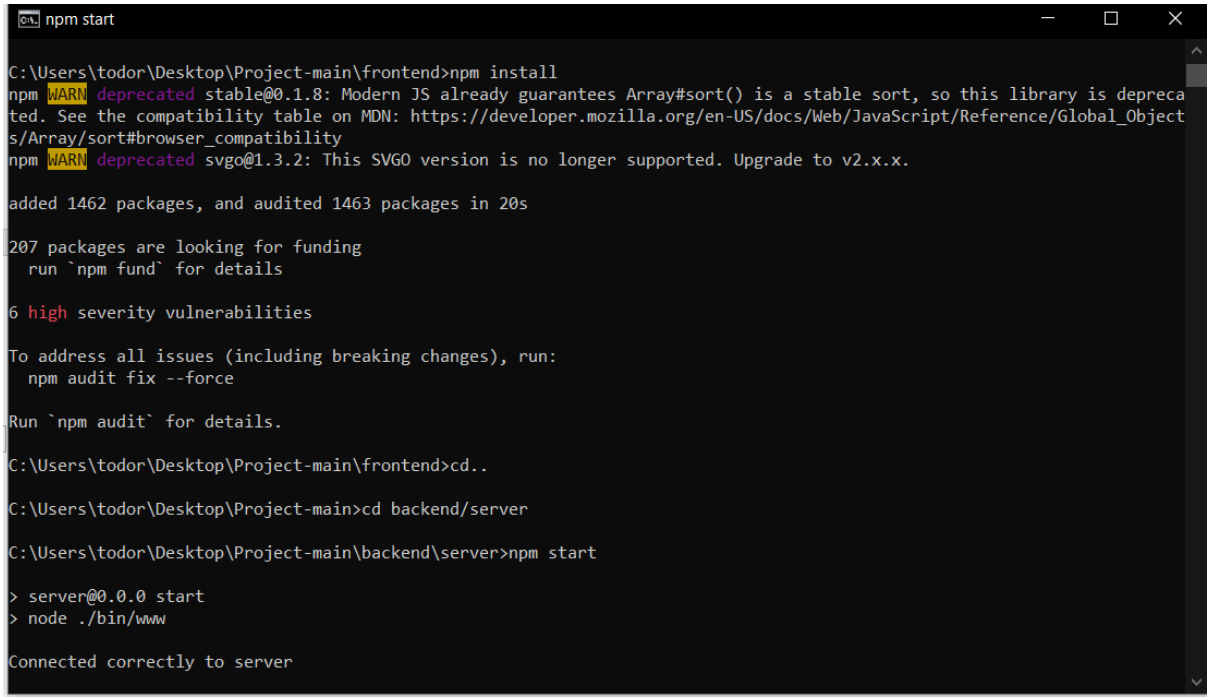
1.) Once you have extracted/cloned the file, go to backend/server and run the command npm install.



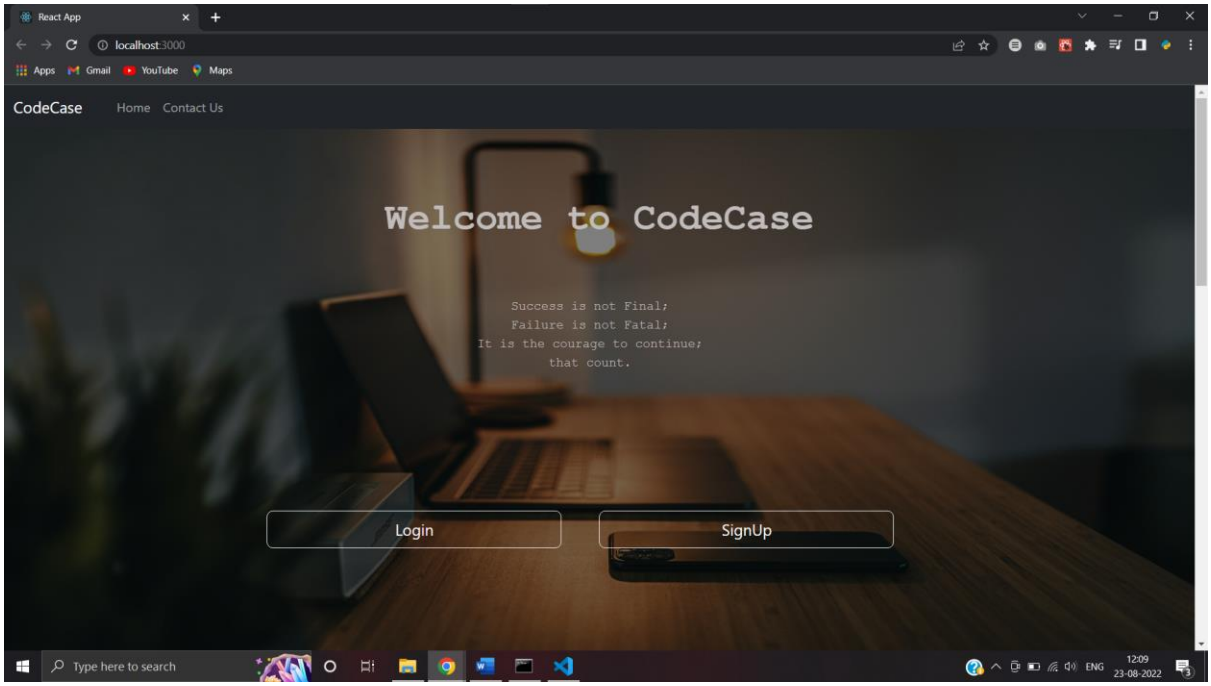
2.) Now go to frontend folder and run npm install. Note you have to run command separately 2 times in sub folders, not in the root directory.



3.) Now go back to backend/server and run “npm start”. Once you see “Connected to server successfully”, server is started.

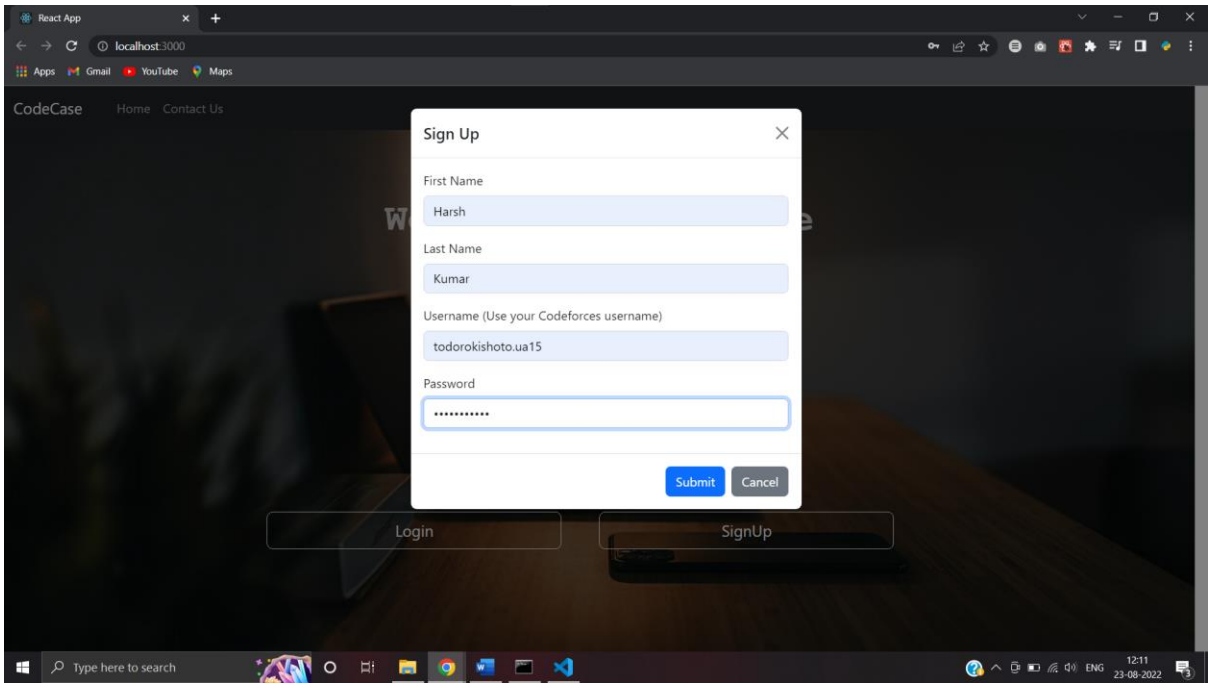


4.) Now, with the server running in background, open one more terminal and go to frontend folder and run “npm start”. Once client is started, you can go to “localhost:3000” to see the web page.



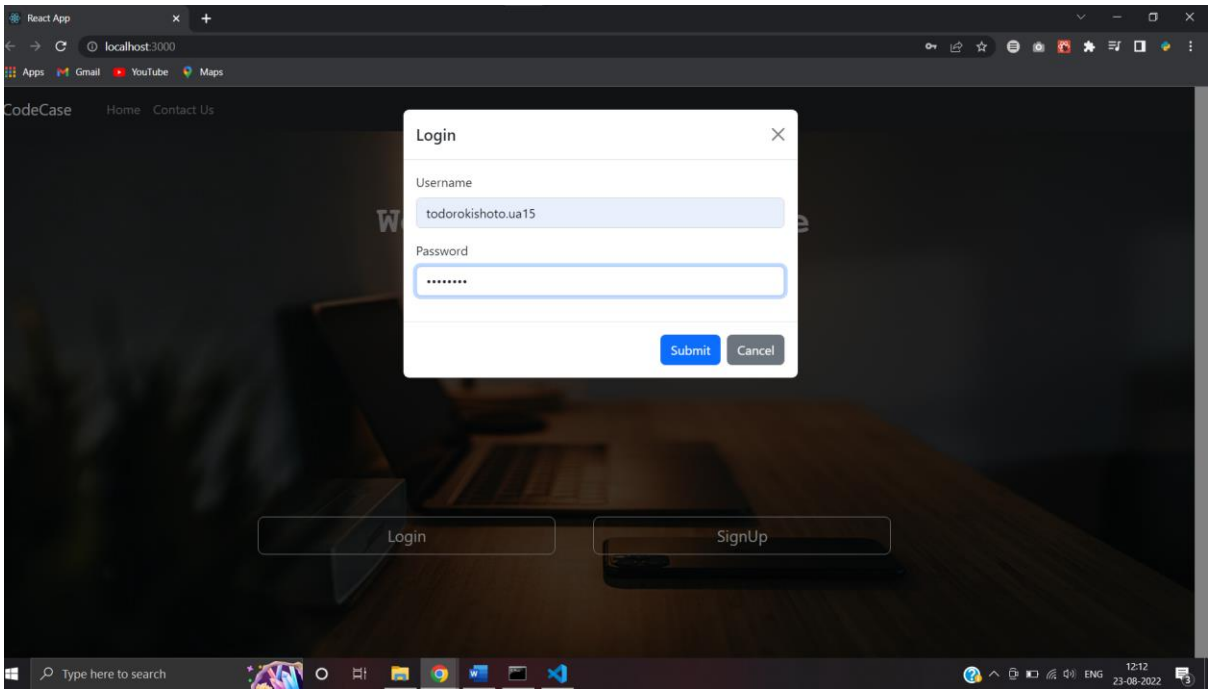
Signing Up.

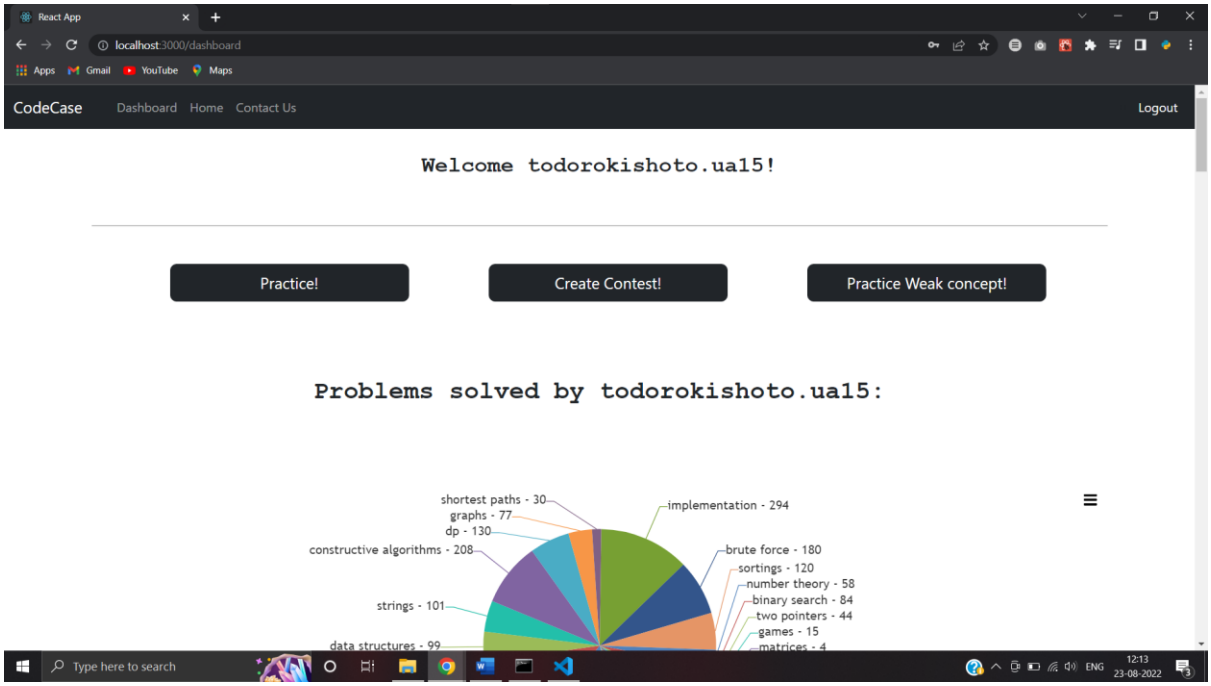
Click on the SignUp button, a popup will appear. Enter the details. In the username field, make sure to put your codeforces username, otherwise it will not function correctly.



Logging in

No click on the login button, enter the same username and password that you used for signing in. After submitting wait for 2-5 seconds and then you will be redirected to dashboard.





Dashboard contains analytics of your solved problems and the problems that you selected in past to practice but did not solve.

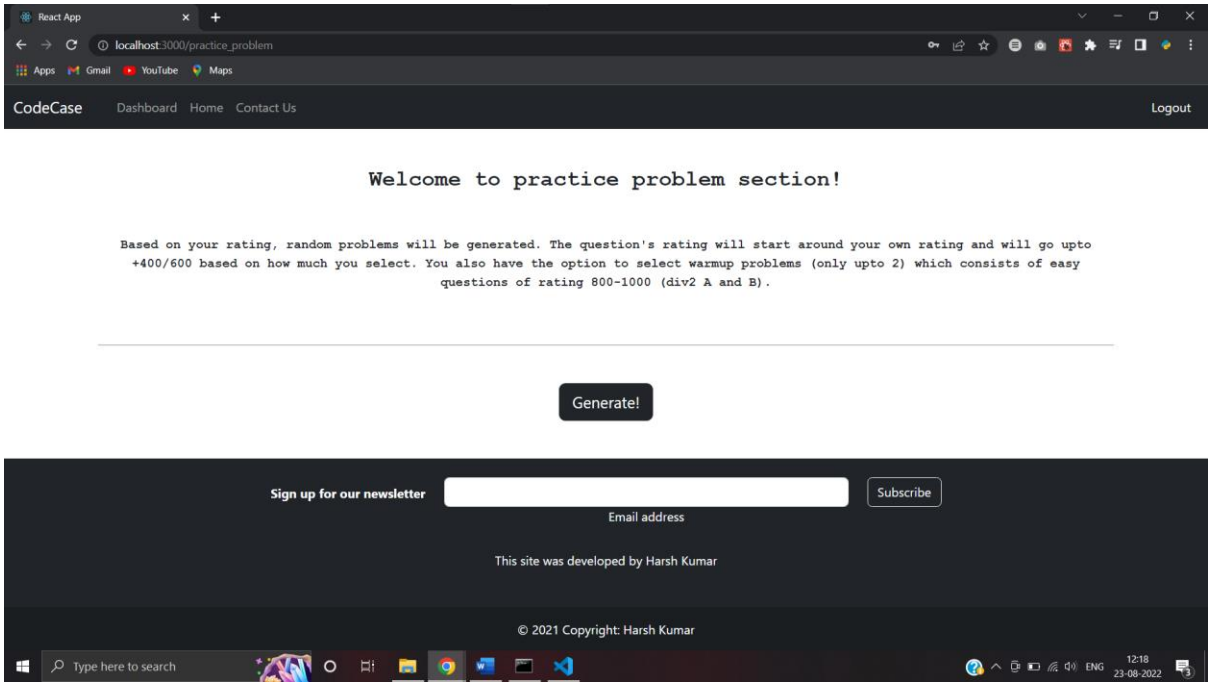
Pending Problems :

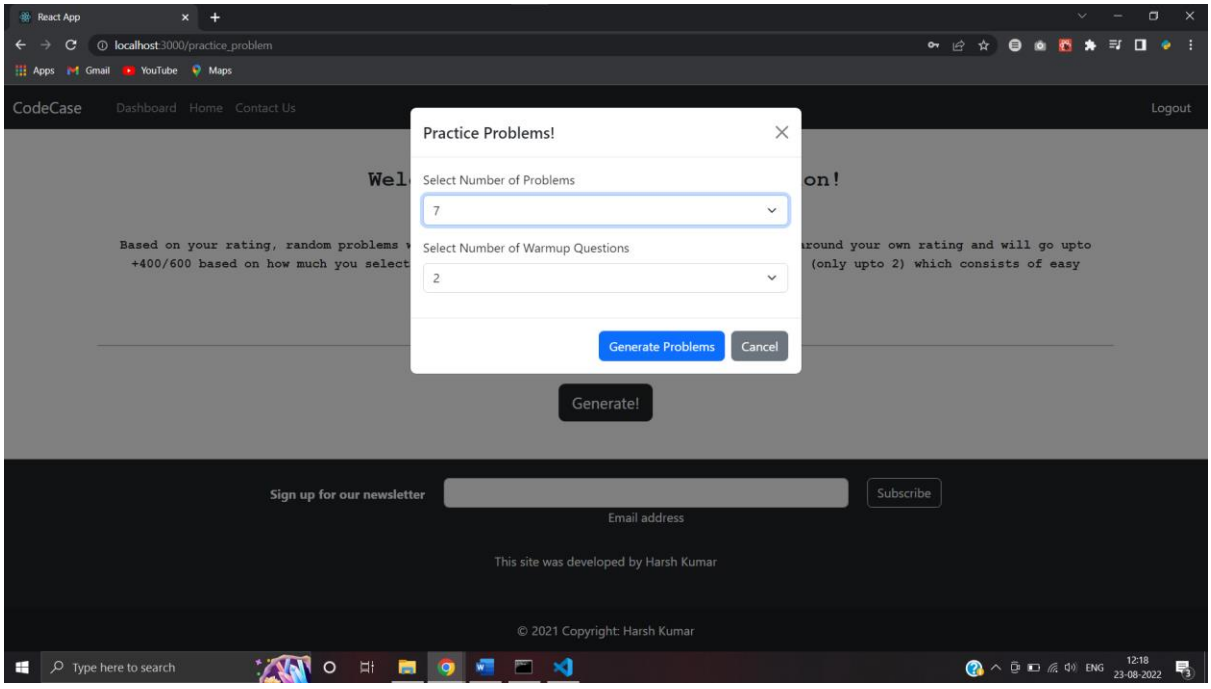
#	Index	Problem Rating	Problem Name	Problem Link
1	A	1500	Lorenzo Von Matterhorn	Go to Problem
2	B	1600	Restaurant	Go to Problem
3	A	1700	BerOS file system	Go to Problem
4	D	1800	Compression	Go to Problem
5	E	1900	Necklace Assembly	Go to Problem
6	C	1500	Pythagorean Triples	Go to Problem
7	B2	1600	Cat Party (Hard Edition)	Go to Problem
8	U3	1700	Block unitary	Go to Problem
9	H	1800	Brevity is Soul of Wit	Go to Problem
10	C	1500	Cut 'em all!	Go to Problem

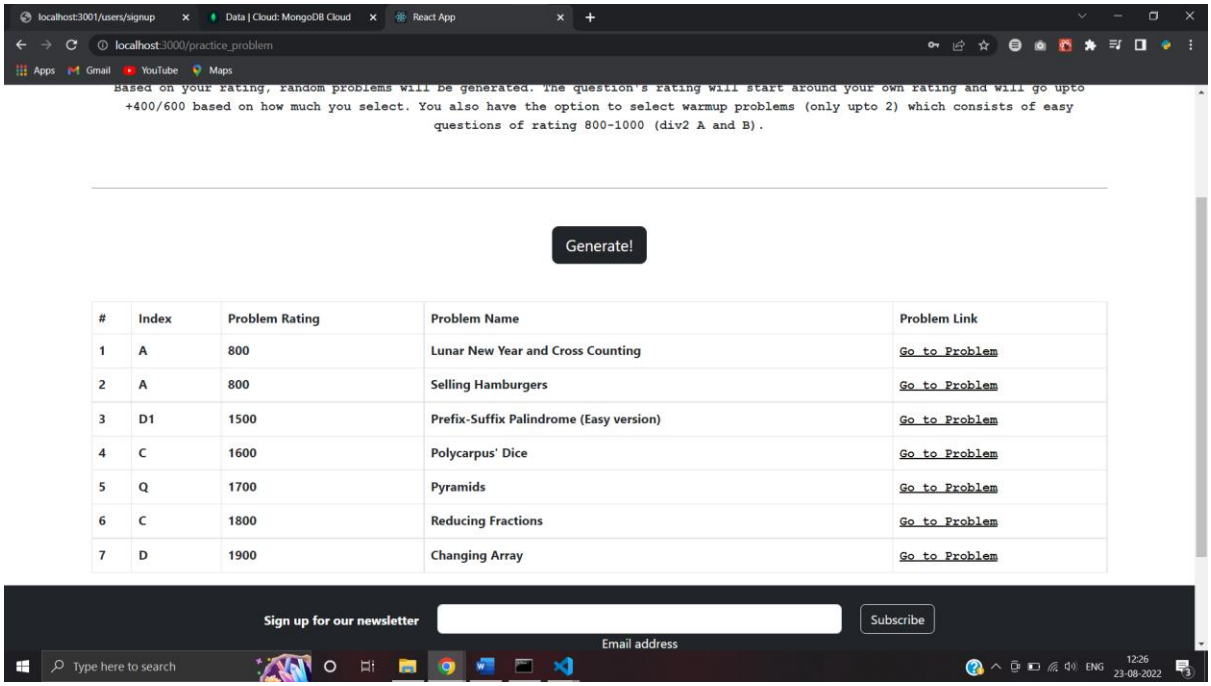
As you go solve the problems, the list reduces. The maximum limit of the list is 50 right now.

Practice Section

In this section, you can select 5-10 problems to practice. The problem generated are based on your rating. It starts from your rating and goes upto +400. In addition, you also have the option to select upto 2 warmup/easy problems which will have difficulty of 200. To generate the problem, click the generate button and enter the problem number.


React App
localhost:3000/practice_problem
Apps
Gmail
YouTube
Maps
CodeCase
Dashboard
Home
Contact Us
Logout
Welcome to practice problem section!
Based on your rating, random problems will be generated. The question's rating will start around your own rating and will go upto +400/600 based on how much you select. You also have the option to select warmup problems (only upto 2) which consists of easy questions of rating 800-1000 (div2 A and B).
Generate!
Sign up for our newsletter
Subscribe
Email address
This site was developed by Harsh Kumar
© 2021 Copyright: Harsh Kumar
Type here to search
ENG
12:18
23-08-2022

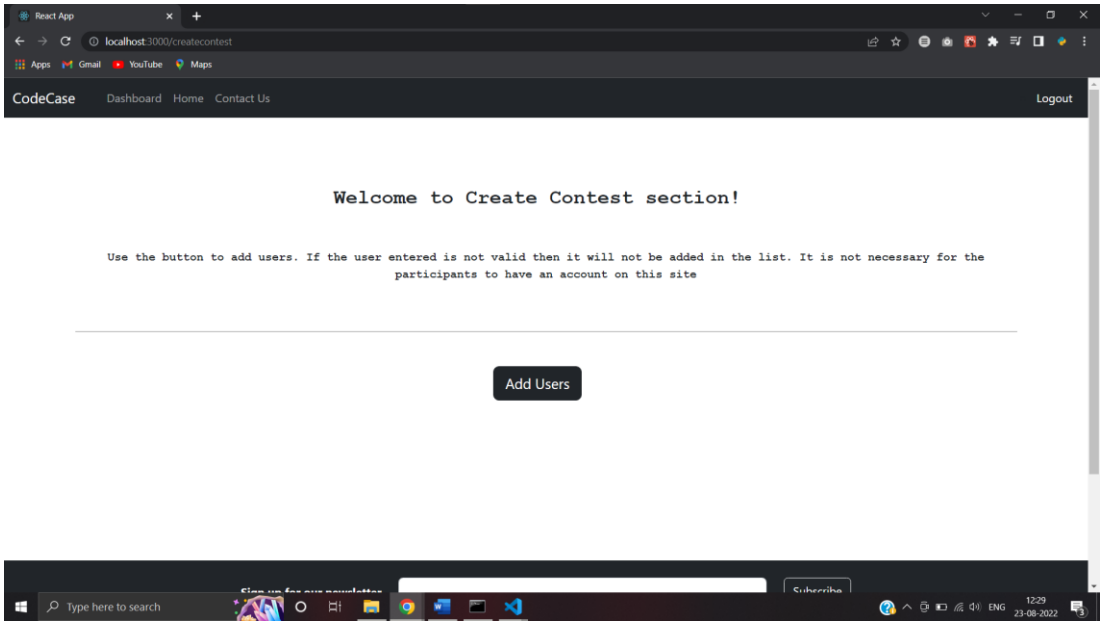

React App
localhost:3000/practice_problem
Apps
Gmail
YouTube
Maps
CodeCase
Dashboard
Home
Contact Us
Logout
Practice Problems!
Select Number of Problems
7
Select Number of Warmup Questions
2
Generate Problems
Cancel
Generate!
Sign up for our newsletter
Subscribe
Email address
This site was developed by Harsh Kumar
© 2021 Copyright: Harsh Kumar
Type here to search
ENG
12:18
23-08-2022

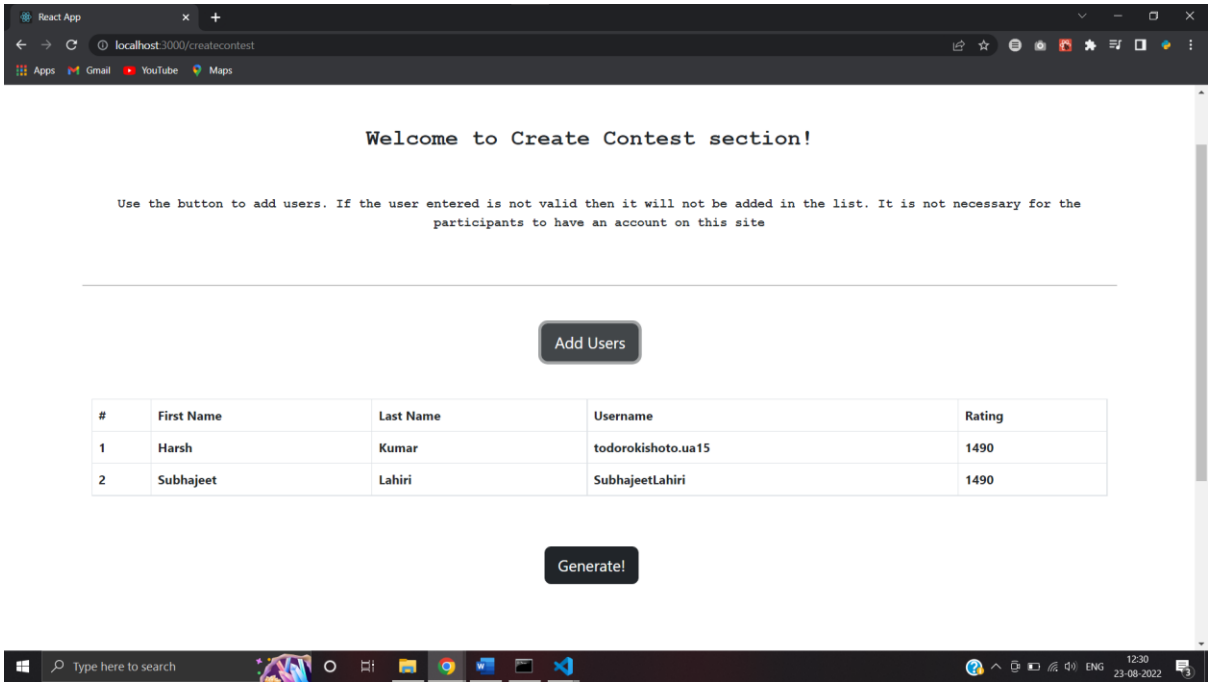


These problem will get automatically added to your dashboard incase if you want to solve them later.

Create Contest

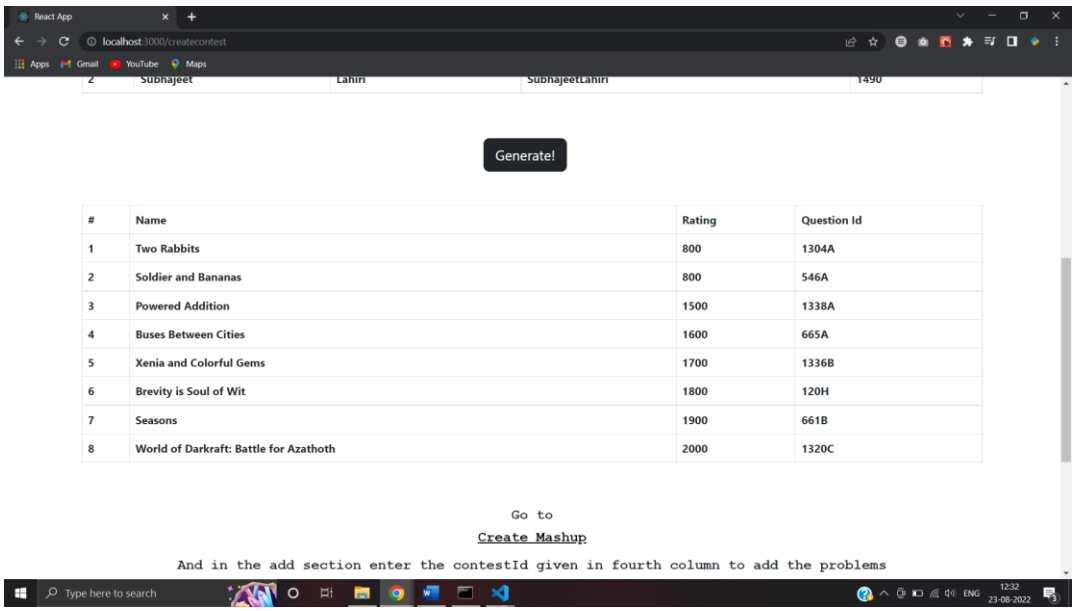
In this section, you can create a mashup contest. You start by adding the participants that are taking part in the contest. It is necessary for them to have an account on this website.





You can use : todorokishoto.ua15 and SubhajeetLahiri usernames for testing.

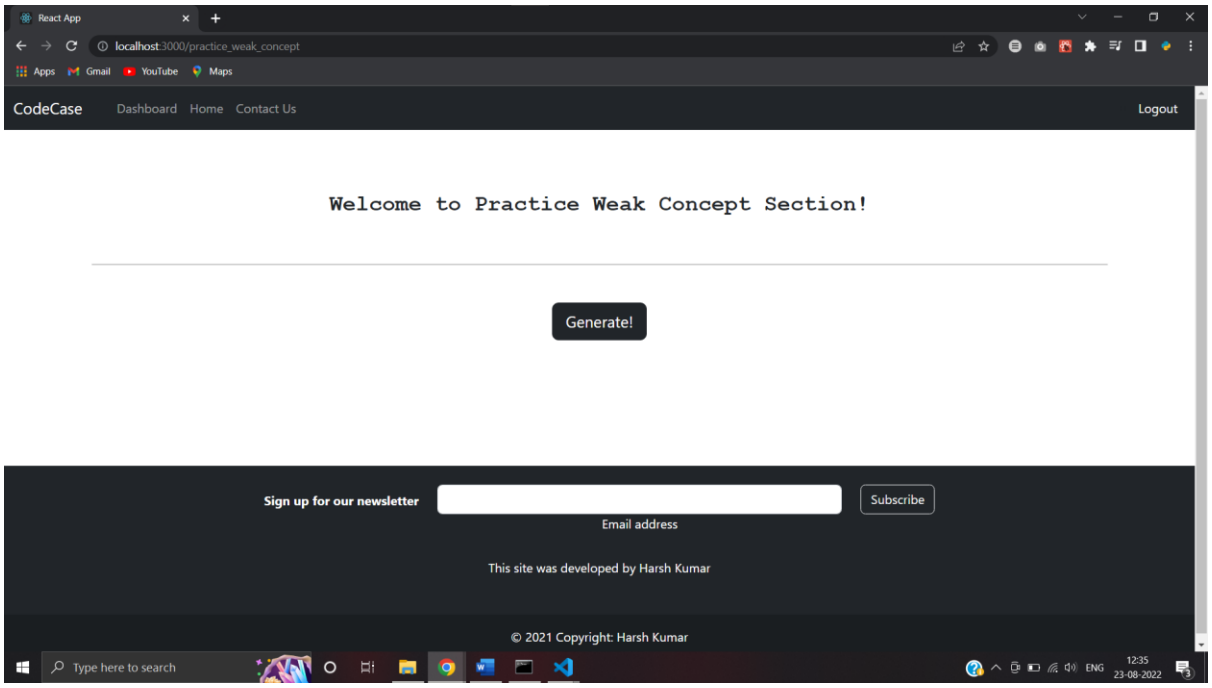
Once users are added, click on generate. Contest created will be based on the rating of all participants so that problem is neither too difficult nor too easy.

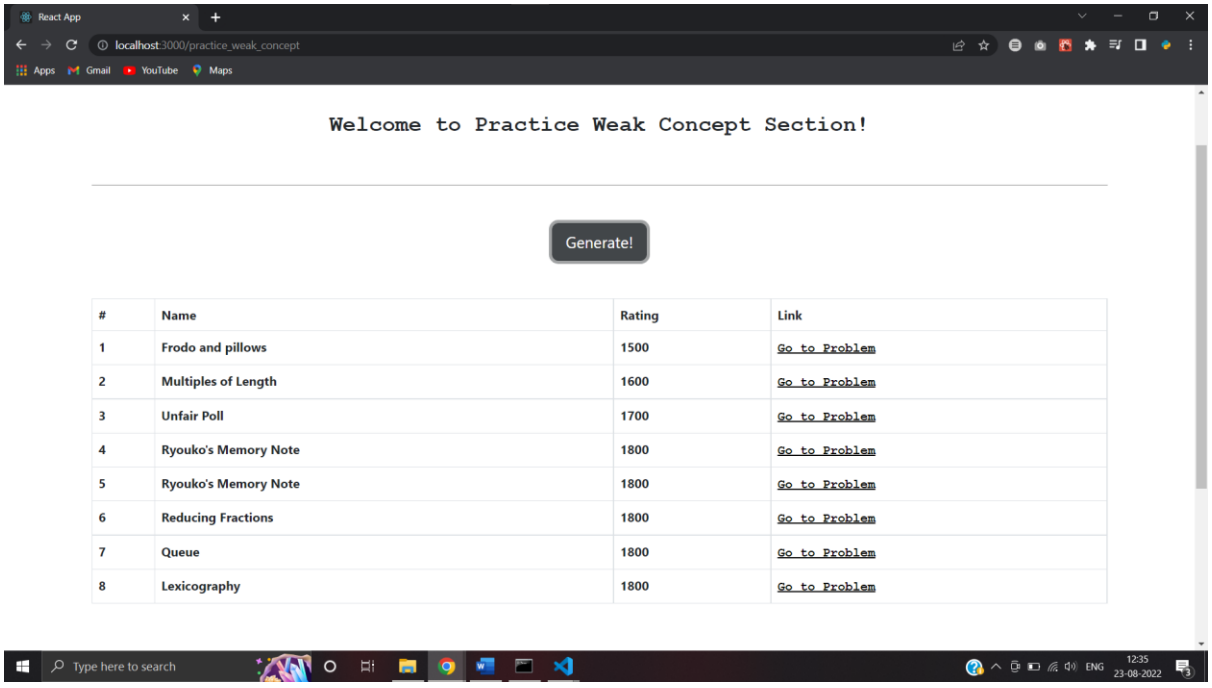


You can now go to Create Mashup and enter the question id generated to create the contest.

Practice Weak Concept

In this section, the generated problem will be based on the concepts you are weak at. The weak concept are decided by the questions unsolved by you. The more you practice, the better the results will be.





#	Name	Rating	Link
1	Frodo and pillows	1500	Go to Problem
2	Multiples of Length	1600	Go to Problem
3	Unfair Poll	1700	Go to Problem
4	Ryouko's Memory Note	1800	Go to Problem
5	Ryouko's Memory Note	1800	Go to Problem
6	Reducing Fractions	1800	Go to Problem
7	Queue	1800	Go to Problem
8	Lexicography	1800	Go to Problem

Problem Faced:

1.) Making the whole website responsive was a huge challenge which took a lot of time.
2.) Post functionality of the fetch was not working as expected, so for post I had to switch to axios.
3.) There were timing related issues, as to when page was loaded and data was displayed. Synchronizing them (upto certain extent) was a challenge.
4.) There were some coding related issues which was solved easily by looking at documentation.
5.) Several bugs were encountered while building backend apis

API Developed:

1.) **localhost:3000/problems** (get) : returns an array with all the pending problem related to an user.
2.) **localhost:3000/problems/upd** (post) : update the problem array related to an user.
3.) **localhost:3000/users/** (get) : returns all user registered on the website
4.) **localhost:3000/users/signup** (post) : registers an user to the website
5.) **localhost:3000/users/login** (post) : logs the user in
6.) **localhost:3000/users/updrating** (post) : updates the codeforces rating of the user.