Online Judge

Documentation

**Judge App**

Models

1. Problem
   1. Name
   2. Description- problem statement
   3. Example- example explanation
   4. Input- example input
   5. Output- example output
   6. Difficulty- problem difficulty tag
2. Test
   1. Problem – foreign key to problem
   2. Input – test input
   3. Output – expected output for input
3. Solution
   1. User – foreign key to user
   2. Problem - foreign key to problem
   3. Language – programming language in which problem is submitted
   4. Code\_file – submission code
   5. Verdict – verdict for the submission
   6. Timsptamp – time of submission

View functions

1. Index- serves to the home page
2. Problems - fetch all the problems from the data base and renders problems.html
3. Problem(id) –
   1. If user is authenticated then fetch the particular problem with given problem id and renders problem.html
   2. If user is not authenticated then redirects to login page
4. Submit(pid) –
   1. process the code submission that is done by user for problem with id=pid
   2. uses helper.runcode and helper.getVerdict
5. result (not in use) – renders submit.html to show the verdict to submission
6. Submissions – show last 10 submissions of user requesting
7. Login\_request – renders to login.html page
8. Login\_check – authenticate the user with given credentials
9. Logout – logout the user
10. Register\_request - renders to register.html page
11. Register\_verify – verify the credentials and create user if everything is ok

**Contest App**

Models

1. Contest
   1. name
   2. description
   3. start\_time
   4. end\_time
2. Scorecard
   1. User – foreign key to the user table
   2. Score – score of particular user in the given contest
   3. Contest – foreign key to the contest table(currently missing)
3. Contest problem
   1. Point – points for the particular problem
   2. Contest – Contest to which the problem belongs

View funtions

1. Index – renders the contest lists that are going to start in the future
2. Contest(id) –
   1. If contest has started then it will render contest\_running.html with and will show the problems.
   2. If contest is not yet started then it will show contest not yet started.
   3. If contest has already been ended then it will show the results(leaderboard)
3. Contest submission – process the code submitted by the user
   1. Calling the API ‘CPP 17 Code Compiler’ to execute the code.
   2. If the code passes the test cases successfully then users score get updated.
   3. Saves the submission with corresponding verdict.
4. Updatescore(user,problem) –
   1. If the correct code is submitted for the first time then adds the score to the user.
   2. If the problem is already solved then score remains same as before.
5. Leaderboard – returns the scorecard sorted by the score in descending order.