



PROJECT REPORT

TEAM ECLIPSE

October 29, 2017

Vighnesh Reddy Konda | Naman Jain | Satti Vamsi Krishna Reddy

Contents

1	Abstract & Proposed Idea	3
1.1	Abstract	3
1.2	Proposed Idea	3
2	Introduction	4
2.1	What's an Online Judge	4
3	Motivation	4
4	Major Softwares/Tools used	5
4.1	Project Development	5
4.2	Project Management	5
4.3	Django Packages	5
5	Requirements & Setup	5
5.1	Installing required softwares	5
5.2	Setup Docker	6
6	Our Implementation	6
6.1	Django	6
6.2	Sandboxing	7
6.3	Ranking System	7
6.4	Ajax	8
6.5	Ace Editor	8
7	Final Deliverables	8
8	Possible Extensions	9
9	Conclusion	9
	Bibliography	10

1 Abstract & Proposed Idea

1.1 Abstract

Eclipse OJ is an online system to test programs in programming contests and also used to practice for such contests. The system can compile and execute code, and test them with pre-constructed data. Submitted code can be run with restrictions, including time limit and memory limit. The output of the code will be captured by the system, and compared with the standard output, the system will then return the result.

1.2 Proposed Idea

We had proposed to implement the following features in Eclipse OJ:

1. Create a website for hosting programming contests and maintaining a database of problems in a categorized fashion.
2. Login for existing users / Registration facility for new users and storing their details in a database.
3. Implement a judge/checker that supports C, C++, Python, Java.
4. Implement a Discussion Forum where users can discuss and post regarding various questions and their solutions.
5. Implement a proper admin portal, where questions can be uploaded and contests hosted.
6. Implement a rating feature, rating will get updated after every contest, a statistical approach to calculate ratings appropriately.
7. Extend the rating feature to leaderboards ranked according to ratings and categorised in various ways (according to say institution).
8. Prevent execution of potential 'malware/untrusted' code that can be uploaded by a user intentionally.
9. Implement a private messaging interface which also supports file-transfers for users to interact and collaborate.*

**To be done if time permits*

2 Introduction

2.1 What's an Online Judge

- An online judge (abbreviated as OJ) is an online system to test programs in programming contests.
- They are also used to practice for such contests.
- The system can compile and execute code, and test them with pre-constructed expected output data.
- Submitted code may be run with restrictions, including time limit, memory limit, security restriction and so on.
- The output of the code will be captured by the system, and compared with the standard output. The system will then return the result.
- Online Judges have rank lists based on user ratings.

3 Motivation

Programming contests are fun but organising one is difficult if the number of participants is large, so it is nice to automate the skill testing in such contests. Also it is good to maintain a collection of all the problems in previous contests so that people can practice. It would be nice if one can submit their code in multiple languages All these can be achieved by having an Online Judge.

Major factors that motivated us to create an Online Judge were:

- Most of the popular online judges today ask for a price to hold a contest.
- No big and famous 'open-source OJs' - so felt the need of one.
- Additionally IIT Bombay doesn't have an online judge of it's own, so we decided to make a state-of-the-art Online judge for our institute, which we strongly believe will increase the community of sport programmers in our insti.
- Team collaboration has been difficult, specifically for learners initially. Messaging systems (if any) already seem to be naive and not user-friendly.

4 Major Softwares/Tools used

4.1 Project Development

- Django 1.11.5
- HTML5, CSS3 (using Sass)
- Javascript (including jQuery, AJAX)
- Docker
- BASH
- MathJax, Bootstrap4 ... etc.

4.2 Project Management

- Git (for Version Control)
- \LaTeX (for Documentation)

4.3 Django Packages

- django-bootstrap4
- django-macros
- django-countries

5 Requirements & Setup

All the instructions mentioned here tested on Ubuntu 16.04 and are applicable to almost all linux distributions with minor modifications. For other OS, kindly install them yourself.

5.1 Installing required softwares

Python is a must for installing Django and we assume that both the latest versions Django 1.11.5 and Python 3.6.3 (at the time of this writing) are installed in your PC. For further instructions refer here[1]

- To install Sass, follow the instructions here[2]
- To install docker, follow the instructions here[3]
- To install django-bootstrap4, run: `pip3 install django-bootstrap4`
- To install django-countries, run: `pip3 install django-countries`
- To install django-macros, run: `pip3 install django-macros`

5.2 Setup Docker

We need to create a container named '*dock_container*' which our OJ uses to sandbox the code before executing them.

- **Step 1:** Move to the Project folder using `cd`
- **Step 2:** Run `docker build -t ubuntu:v1 'path to project folder'/Dockerfile`
- **Step 3:** Run `docker create -id ubuntu:v1 --name dock_container`

Note: You may need to start the dock_container everytime you restart the PC. You may automate this process easily however.

Yayy! And thats it, we are ready to run our project!

6 Our Implementation

6.1 Django

- All the frontend UI for our website is done using HTML, CSS, JS , Bootstrap.
- All the various modules like login, registration, discussion forum, database of questions, leaderboard, messaging ...etc are done as 'separate apps' thus accounting for reusability.
- We would implement a basic login/registration page with Django, using Django's authentication packages at 'django.contrib.auth'
- We plan to use Django's argon2 hashing algorithm to provide secure hashing for passwords present at (django.contrib.auth.hashers)
- Customize the 'admin.py' in Django specifically for adding problems and hosting contests in a simple way using sqlite3 databases.

- All the submission requests will be handled *via*. a queue, hence preventing any possible server crash.
- Every code is sandboxed using Docker to avoid any harm done by running untrusted/malicious codes.
- Implement a Rating system that reflects the user's performance based on a simplified Elo Rating System.

All user start with 1500 rating. The probability of a user with a rating R_A being placed higher than a person with a rating R_B is equal to:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \cdot \quad [4]$$

6.2 Sandboxing

We intend to prevent any harm to server caused by execution of 'untrusted code'. This is commonly called 'Sandboxing' in Computer Security. At first we chose to implement a Chroot Jail but we found using Docker was much simpler and elegant So, we decided to use Docker [6][7]

All the users submitted codes are executed in a container called dock_container. We copy the testcase folder into the container and do the testing. So if a user sends any malicious code if at all something happens the container gets effected rather than our main system

6.3 Ranking System

Our rating system is similiar to Elo rating system. the calculations part is simplified as below:

$$PerformanceRating = \frac{total\ of\ opponen\ ratings + 400 \times (Wins - Losses)}{Games}$$

- The Elo rating system is a method for calculating the relative skill levels of players in multiplayer type of games.
- We found that Elo rating system is most appropriate for us and also Codechef and Hackerrank implement their rating system mainly based on Elo (though not entirely the same)

Most of the popular OJs do not keep their rating mechanisms open and the number of factors the rating depends on is huge

6.4 Ajax

We have used Ajax JsonResponse method to create an Asynchronous HTTP request in the signp form. Once the username input box loses focus, ajax sends an ajax request to server containing the preferred username by the user. This is checked with the existing user database and if there exists such a user, the user is notified dynamically.

6.5 Ace Editor

We linked the editor which the user can directly use to code. He can check with his own custom testcases. He can directly submit his code to a problem. The text editor supports C,C++,Python2,Python3 and Java with syntax highlighting. The editor can be customized by changing its theme. We provide one Dark and one Light Theme

7 Final Deliverables

We have successfully implemented all the features we mentioned in the project proposal.

1. Create a website for hosting programming contests and maintaining a database of problems in a categorized fashion.
2. Login for existing users / Registration facility for new users and storing their details in a database.
3. Implement a judge/checker that supports C, C++, Python2, Python3, Java.
4. Implement a Discussion Forum where users can discuss and post regarding various questions and their solutions.
5. Implement a proper admin portal, where questions can be uploaded and contests hosted.
6. Implement a rating feature, rating will get updated after every contest, a statistical approach to calculate ratings appropriately.
7. Extend the rating feature to leaderboards ranked according to ratings and categorised in various ways (according to say institution).
8. Prevent execution of potential 'malware/untrusted' code that can be uploaded by a user intentionally using Docker.
9. Linked Ajax Ace editor where user can directly code and check with his custom inputs or submit for a problem

8 Possible Extensions

- Messaging System with support for group chats
- Automatic Test Case generation for problems depending on the constraints
- Virtual contests where a user can take a past contest like a current contest though his rating won't get updated
- Better rating system that takes into the consideration of the problem weights

9 Conclusion

We learned the main features of Django, Sandboxing using Docker, database relationships using sqlite3, and improved our knowledge of collaboration with Git and how open source projects are really developed, Bash

References

- [1] Link: <https://docs.djangoproject.com/en/1.11/topics/install/>
- [2] Link: <http://sass-lang.com/install>
- [3] Link: <https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>
- [4] Link: http://en.wikipedia.org/wiki/Elo_rating_system#Mathematical_details
- [5] Link: <https://docs.djangoproject.com/en/1.11/>
- [6] Link: <https://blog.remoteinterview.io/how-we-used-docker-to-compile-and-run-untrusted-code-2fafbffe2ad5>
- [7] Link: <https://stackoverflow.com/questions/1014505/how-does-the-acm-icpc-online-judge-prevent-malicious-attacks>
- [8] Link: <https://github.com/zostera/django-bootstrap4>
- [9] Link: <http://flipclockjs.com/>
- [10] Link: <https://stackoverflow.com/questions/46867211/django-template-custom-form-using-text-area-not-working>
- [11] Link: <https://ace.c9.io/>
- [12] Link: <https://djangopackages.org/grids/g/countries/>
- [13] Link: <https://pypi.python.org/pypi/django-macros/0.3>
- [14] Link: <https://getbootstrap.com/>
- [15] Link: <https://stackoverflow.com/questions/46866071/javascript-fill-input-with-value-containing-multiple-lines>
- [16] Link: <https://stackoverflow.com/questions/46862684/downloading-input-from-textarea-to-server-as-a-file-django>
- [17] Link: <https://stackoverflow.com/questions/46823308/multithreading-multiprocessing-in-django>
- [18] Link: <https://stackoverflow.com/questions/19033818/how-to-call-a-function-on-a-running-python-thread>
- [19] Link: <https://stackoverflow.com/questions/6862250/change-a-django-form-field-to-a-hidden-field>
- [20] Link: <https://simpleisbetterthancomplex.com/tutorial/2016/08/29/how-to-work-with-ajax-request-with-django.html>
- [21] Link: <https://stackoverflow.com/questions/19123715/django-for-loop-to-iterate-form-fields>
- [22] Link: https://www.w3schools.com/jsref/event_onload.asp
- [23] Link: <https://www.htmlgoodies.com/beyond/javascript/article.php/3724571/Using-Multiple-JavaScript-Onload-Functions.htm>
- [24] Link: <https://stackoverflow.com/questions/18497688/run-a-docker-image-as-a-container>
- [25] Link: <https://stackoverflow.com/questions/22383547/bootstrap-dropdown-menu-is-not-working>
- [26] Link: https://www.w3schools.com/jquery/ajax_getjson.asp