TEST Engineering & Management

January-February 2020
ISSN: 0193-4120 Page No. 16058 - 16062

Enhanced Progressive Library of Algorithmic Content and Optimized Performance Tracking in Programming

Abhishek Yadav, Computer Science Department, Chandigarh University
Somesh Awasthi, Computer Science Department, Chandigarh University
Shubham Sharma, Computer Science Department, Chandigarh University
Ankit Verma, Computer Science Department, Chandigarh University
Tanu Dhiman, Computer Science Department, Chandigarh University
Surabhi Kaul, Computer Science Department, Chandigarh University
Gulbadan Khehra, Computer Science Department, Chandigarh University

Article Info
Volume 82
Page Number: 16058 - 16062
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 28 February 2020*

Abstract:
Progress in computer technology, copulated with the burgeoning need for computer professionals is producing new paradigms for education and training. Participants, from the outlook of these educative and training paradigms, expect a rich preparation environment backed by well-composed resources. This application aims to provide on-demand, anytime/anywhere, high-quality instruction with good community support. To remain viable in the realm of online learning the application uses novel system architecture to perform efficiently and flexibly. The application will work on the policy of distributed learning, which enables the learner, tutor and the content to be placed in several, non-centralized sections so that learning and instruction can occur independently of moment and area

Keywords: *computer technology, distributed learning*

1. INTRODUCTION

A wide variety of articles and posts are available on the internet about various data structures and algorithms but they are all scattered and not available at one common place where a user can access them. The availability of a place where all these resources are organized in a good way can prove to be of great use. Also, the scarcity of resources which help the user visualize what they are learning has resulted in users not able to develop deep knowledge in algorithms and data structures thus, there lies a great scope to fulfill this need and the project aims to do the same. Algo-Box is aimed to be a utility tool that enhances the learning experience for people in the programming world with chief focus on centralizing different resources at a single place. Resources may include Library of well explained Algorithms and Data Structures which brings up all the Algorithms and Data Structures from basic to advanced level along with well explained form and usable code, An open community for all It's a community of learners who help each other out as well as contribute back by improving the content for everyone. A community not only helps you when you're stuck, it also keeps you accountable. Learning is more fun when done together, Ever improving content , Every piece of content is a wiki, which means that anyone can improve it. This ensures that the article you are reading is always up to date, and what you write today will keep improving over time, Providing a coding calendar so that you never miss out any Programming contest or hiring challenge and providing a Problem Classifier which recommends problem to the users related to the topic they are

January-February 2020
ISSN: 0193-4120 Page No. 16058 - 16062

weak at. Systems like this have been implemented before but with not as much functionality and support for the community and this form for practicing problems on different topics for all around development.

Having a platform that can be used to track progress and learn the required material from it as well will be a huge help to people who usually spend a lot of time on searching material and not actually comprehending the knowledge. Providing a common platform will be beneficial for everyone.

A very important feature of any web app is its performance, fetching data from the database and also writing data to it. But it is also a very time and memory consuming operation to read from and write to a database, especially when the database is online which may slow the process even more as the internet connection might not always be good. A significant way to improve this would be to implement a server side caching mechanism as our data is not very dynamic and requires change only when either the database will be updated with new content i.e. new blogs or new contests. We can create an API for caching the data from the database and do periodic fetching from the database and cache the data after every fetch. The API will provide direct data to the web application so that we do not have to fetch data from databases everytime and this will speed up the fetch and response procedure and increase the performance.

2. RELATEDWORK

Various websites and open source projects have aimed at gathering all the good resources and trying to place them together, but the quality of the resources degrade overtime because the moderators are not able to handle and continuously do a quality check for all the available resources. This project eliminates the major work that themoderators previously had to do by involving the community and allowing them to moderate and update the resources while keeping in check the quality of those updated resources. Also the previously developed websites did not have a checkpoint/tracking system that allows them to track their progress which can allow them to assess how much they have progressed over time. Some websites such as Code-Drills have provided a recommendation system to recommend problems to users from their weaker areas but because no updates were made to the platform, the pool from which the problems get recommended is very small and that does not provide good recommendations over time even after more and more problems can be easily added for multiple topics. All the platforms that have been built do not have any feature for teams and to accommodate the progress of a team as a whole, but providing that feature can be a huge boon for teams that aim at doing good at team contests such as ICPC, SnackDown and many more. A feature for tracking team progress would be a very good feature but has not been implemented yet.

[1]Platforms with a similar vision also exist, Common Lounge is the closest example of a website that provides an interface and functionality similar to ours, but it is mainly aimed at high school students that seek to qualify for and earn a medal in the Internal Olympiad of Informatics (IOI) so the preparation material and the website is moulded in a fashion similar to the one required by the Olympiad of Informatics, one of the additional features that Common Lounge had was that they provided stepwise learning and content unlocked after some series of educational material followed, but what they lack is the problem prediction and progress tracking both overall and also for some particular topic.

3. PROPOSEDWORK

One of the major problems that applications of this type face are that they suffer from latency due to calls to database and other external API's. This

January-February 2020
ISSN: 0193-4120 Page No. 16058 - 16062

problem causes the application to become very slow and degrades the performance by a very large margin. To remove this is a major challenge. The two most used components of our web app are the Blogs app and the Calendar app. The Blogs app has to do a database query every time a user tries to see the blogs and the CalenderAPI has to be called every time that the user opens the homepage or the Calendar app. Database queries are sometimes slow when a huge amount of Blogs get added which would affect our ability to scale and similarly the Calendar API calls are slow depending on the API provider.

To eliminate this problem we proposed to create two separate apps within the project and our framework Django has a great provision for it. We will mainly try to implement Server side caching and user side caching at every possible step to reduce the time for each query. The Blogs application and the Calendar application will be hidden directly from the user but we will create an API system for posting and requesting blogs from the database where we can easily cache the files on server and render without much latency. [2]We will also introduce a Karmasystemwhichwould be a reputation system for the users which would make it easy to identify and allow the community to recognize the top contributors and this system would allow for easy contributions and the people with higher reputation would be eligible to rate articles which would in turn allow the entire website to function in a better flow.

We will be using a python based framework known as django for our backend services and we will be using javascript frameworks and html, css for the frontend. Django is a great choice for projects like this as django is a framework whose very existence is complemented by the performance it provides, also django allows the developers to perform some good amount of operations and run system apps along with the web application on the host server / computer. The system apps and web servers we will be using can easily be integrated with django and give a better performance. System apps and modules that we require for doing the server side caching that we proposed can also be integrated with django and django has inbuilt support for thesemodules.

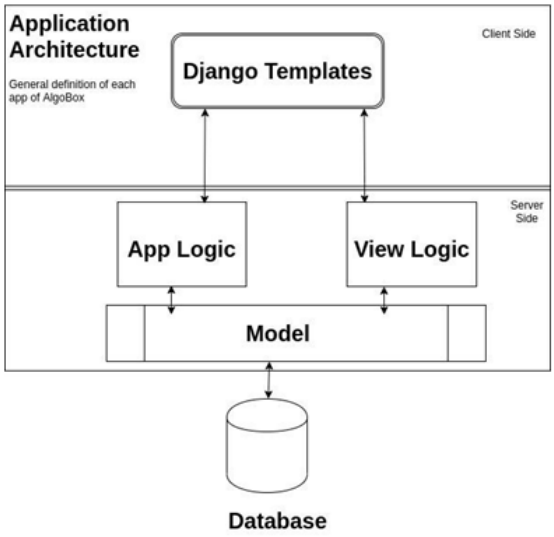

Figure 1: Single Application Architecture

The implementation model proposed is that instead of fetching data every time a user would request, we would cache the data as much as possible and only update data when some event is triggered or after some fixed interval of time. For deployment we will be using NGINX web servers as they are one of the best web servers to be used with a django backend as they seamlessly integrate with each other and complement each othersperformance.

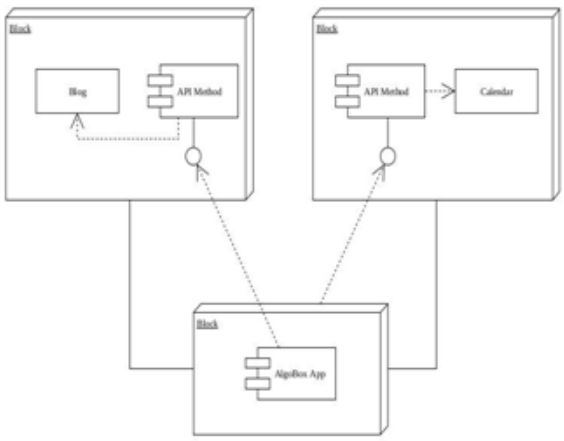

Figure 2: UML Diagram

January-February 2020
ISSN: 0193-4120 Page No. 16058 - 16062

Also, the project uses Microservice architecture that structures the application as a group of small independent services, modelled around the main application. Each of these services is self-contained and implements a single feature.

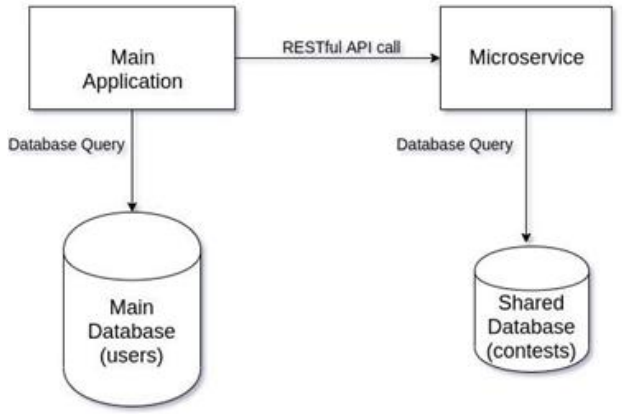

Figure 3: System design for MicroserviceI.

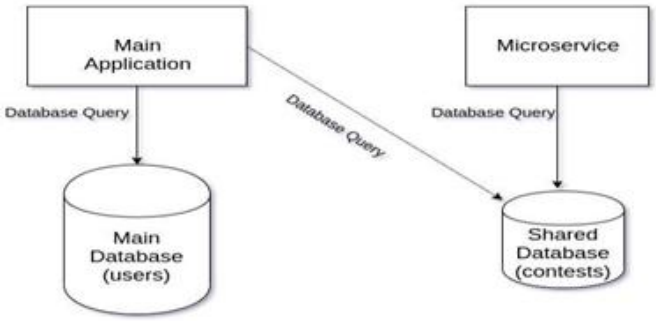

Figure 4: System design for MicroserviceII.

This architecture makes the application easy to scale and maintain, as different services are independently deployableand scalable, each service also provides a firm module boundary, even allowing for different services to be written in different programming languages and can be operated by separate teams. The design also leaves the system with a better fault isolation mechanism; if one microservice fails, the others will continue to work. Microservices clarify security monitoring because the various parts of an app are detached. A security predicament could happen in one section without affecting other areas of a particular project.

4. RESULT

The applications were implemented and integrated with the help of django framework from python. The Blog application has all the CRUD (Create, Read, Update and Delete) operations implemented which allows users to add blogs to post articles that will be public and visible to everyone, we have also tried to introduce a karma / reputation system so that not everyone in able to post blog posts publicly that would degrade the quality of articles as all public articles are never of the best quality. The karma / reputation system will be determined by the community of users for the application, users with low reputation will also be able to post articles but before being published or being publicly available, they would have to go through review from people who have higher reputation system in the community and based on the quality of the article / blog written by the user, they will be provided with more reputation depending on how much their idea / article/ blog gets hits among the community, this would ensure a good flow of resources without stagnating or putting the role of publishing on one team. The Calendar API has also been successfully integrated into the app which itself works on an API which is publicly available on demand from Clist. The user problem recommendation system is integrated as a crucial part of the user application interface, we have tried to used the best possible heuristic approach for recommending problems based on the type and form the user wants to solve problems from, various websites like InterviewBit, LeetCode, mettl focus only on interviews and not competitive programming in general, so questions of there type had to be separated for different pool of users, as not all users have the same goal for using the web application, it is completely upon the interest of the user to choose which side he would like to solve problems from. Also, an additional teams feature has been added for teams that are preparing for contests like ICPC, SnackDown, VKCup and other major programming competitions being held, this uses

January-February 2020
ISSN: 0193-4120 Page No. 16058 - 16062

collective user data for predicting problems to be solved, the recommendation system also strictly follows the standard of problem rating which consists of taking data of how many users were able to solve the problem, the average attempts required to solve the problem and also the total number of submissions, this tries to ensure the best prediction pattern allowing users to get more value out of their visit to the website other than just reading the algorithms / data-structures they will be able to find problems and also techniques that will help them master thosetopics.

The performance part of the application was a major part of the project and that is what was given the highest priority in the projec. To ensure better performance, we have tried to implement as much of server side caching which could be possible on the available servers. Even after we were able to make the database queries faster and improving our queries as well as abstracting our data and converting some major database centric apps into API's, we additionally have used NGINX servers which are quite famous for their ability to provide developers the power to code and redirect their micro-routines and load balancing, in addition we tried to review other services like apache-tomcat and also wgsi but the degree of customizability available in NGINX was unmatched.

5. CONCLUSION

The application was successfully implemented as per planned and deployed for public use. To get the initial traction to the website a group of selected individuals were invited to try the application and give their feedback. From the feedback received, it was deduced that users were able to easily use all the CRUD operations. The application yields a very good response time and is equally compatible with various devices with diverse aspect ratios. Also, it was noted that the integrated features like the Coding-Calendar and the submission fetcher were functioning very optimally due to their unconventional server side caching mechanism. The performance tracking features successfully enabled the users to keep an eye on their recent practice sessions and revealed the parts that the user should work more on. With all the functionalities performing at par with what the team expected the website will be rolled out with blogs from selected topics within 5 - 7days.

6. ACKNOWLEDGEMENT

We owe a very sincere thanks to Chandigarh University for providing us the platform and support to implement this project "Enhanced progressive library of algorithmic content and optimized performance tracking in programming". Also, we would like to thank our project mentor Ms. CharanpreetKaur for instigating within the requirements of the research and providing us the opportunity to implement theidea.

Also, we would like to thank Dr. Ruchika Gupta, for guiding us and providing us with her valuable feedback which helped the team to improve different aspects of the project and make it more usable. Her weekly assessment of the project led us to build a robust and accurate system in compliance to the requirements of theproject.

REFERENCES

[1] Maryam Alavi, George M. Marakas, YoungjinYoo.(1 Dec 2002). A Comparative Study of Distributed Learning Environments on LearningOutcomes.

[2] P. Chart and S. Stolfo. Experiments on multistrategy learning by meta-leaming. Submitted to CIKM 93,1993.

[3] Chen, Bill & Bottoms, Bill & Armstrong, Dave &Isobayashi, Atsunobu. (2015). ITRS 2.0: Heterogeneous Integration. Solid State Technology. 58. 13-17.

[4] Choe, Jeongdong. (2015). Comparison of 1Ynm NAND architectures and beyond. Solid State Technology. 58. 24-25.

[5] Vogler, Debra. (2013). A Preview of Semicon West 2013. Solid State Technology. 56. 30-32.