Digitomize: A Centralised Platform for Competitive Programming and Portfolio Management

Sahil Patil[1], Sanket Ravan[2], Swapnil Parte[3], Nitin Gadkari[4]
[1,2,3,4] *Department of Data Science D Y Patil College of Engineering and Technology, Kolhapur, India*

***Abstract*—As competitive programming becomes increasingly popular among students and software engineers; there is a demand for an application that could help the user gain clarity on the status of his/her learning journey. Currently, users participate in various contests and use several web applications, such as LeetCode, Codeforces, CodeChef, and HackerRank. In doing so, users have to jump from one platform to another, making it difficult for them to assess their progress and performance as software engineers.**

Digitomize is an attempt to solve this problem. It does not replace all other platforms used in the competitive programming community; instead, Digitomize gathers all information regarding contest schedules, performance, etc., from every platform used in the software development industry in one place, saving the user's time significantly. The user receives notifications about upcoming contests and the progression of his/her career with the help of graphical representation. Also, the user can consolidate a portfolio that consists of his/her codes in all platforms in use.

Digitomize is based on the most recent tech stack that consists of React.js on the frontend, Node.js with Express.js on the backend, and MongoDB as the database layer. Due to these technologies, it is possible to ensure fast and scalable

I. INTRODUCTION

As time has gone on, competitive programming has evolved from merely being entertaining into becoming a large part of the process of skill acquisition. More and more engineering students/engineers find themselves solving algorithms for fun, learning how to think rationally, and improving their coding abilities, as well as keeping them in shape in anticipation of technical recruitment procedures. The development in that direction was brought about largely by websites like LeetCode, Codeforces, CodeChef, and HackerRank, which have made this type of activity more easily accessible via a beginner-level set of problems and up to rated contests requiring high concentration.

Despite the fact that there are plenty of places for practice available, none of the websites communicate in real-time with each other. Therefore, using three different websites to practice competitive programming will force users to monitor their results using three linear dashboards. Often, they won't receive notifications regarding upcoming contests, and while working on improving their rating on one website, they'll fail to notice that their performance improved on another. Lastly, switching between accounts will cause unnecessary stress, which over time may diminish the efficiency of their practice.

The professionalism of the problem at hand gets further complicated by another problem. For those who want to be employed, it would be nice if they get a chance to showcase their credentials as far as programming is concerned. But if you just throw some random links to different profiles, which may be on different platforms, it is unlikely that your programming capabilities will shine. Thus, there is a need for programmers to be able to showcase their journey as a programmer in one place that will make sense to them and to the person evaluating their skills.

To solve these problems, Digitomize was designed to collect data from programmers across various platforms using Application Programming Interfaces (APIs). Thus, you would be able to view all the information about your career as a programmer in one place. Also, you will get a portfolio, which will dynamically keep updating as you compete and solve different problems, while offering personalized insights about your performance. This way, you can

concentrate on sharpening your skills without worrying about logistics.

II. LITERATURE REVIEW

The growing interest in online judges and competitions has revolutionized the approach towards teaching competitive programming. Research proves that individuals who have been involved in a structured problem-solving learning environment perform better during technical interviews and create efficient software for practical use. Nonetheless, several studies reveal that the segmentation of these sites discourages individuals because they have to collect all the required data, such as statistics, history, and solutions, from different sites and manage their own collection of collected data. Consequently, they lose motivation and cease participation in the activity.

Principles established by Robert C. Martin for designing software that enables maintenance and extensibility can definitely be applied in the case of Digitomize. In cases where a software solution requires integration of one or multiple 3rd party APIs in order to enhance its performance, it should be designed in a manner that makes it possible to adapt to the new changes brought about by the API. This means that a tightly coupled software architecture is highly vulnerable to being rendered useless due to any change in the API.

Yet another approach that can be applied to building up the platform architecture is the application of design patterns. To give an example, the Adapter pattern might be used for transforming different outputs from different solutions into one uniform output for the platform. The Observer pattern can assist with building up a notification system, so that users will always know whenever there is an update on the platform, without any need to look for updates by themselves.

MongoDB has been selected as the underlying data storage layer because of its successful past experience as a data storage medium for flexible and different shapes and frequencies of changes in data. The fact is that the contest data details, user desired usage types, and previous user performance have all very different shapes and frequencies of updates. As a result, using a document-oriented data storage would make the system work much better compared to the rigid data model.

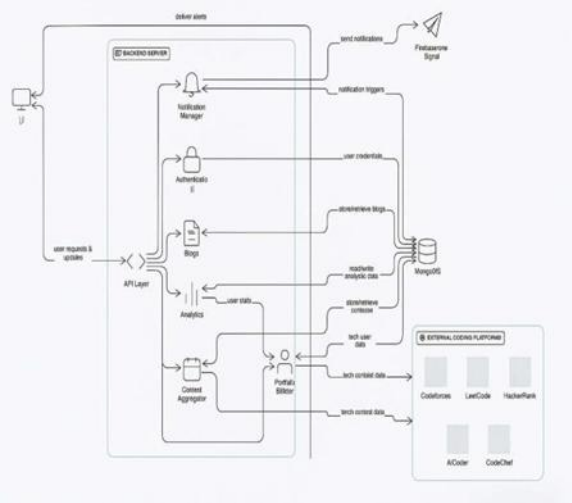
The research in the field of gamification has also helped in understanding the key principles of how to ensure user engagement in learning systems. Gamification features such as progress bar, maintaining streaks, and achievement badges are found to increase user participation in the system. This research finding was applied by Digitomize in creating visual representations of user achievements and in developing automated reminders as a means to ensure continuous actions.

III. METHODOLOGY

Digitomize is a product that has undergone development through modular methodology in order to have a scalable and adaptable framework for growth by adding functional capabilities. The approach that has been adopted in the development of Digitomize does not consist of having it as one standalone application, but rather the creation of Digitomize through the separation of the project into four major components that form Digitomize's hierarchical architecture. These four components include the user interface, server-side processing engine, data repository, and an external application programming interface (API) component. Each component has its own specific responsibility, while interactions between different components follow certain protocols.

A. System Architecture

In essence, Digitomize consists of a client server model whereby the application has a React.js front end as the interface where users interact with Digitomize. The Node.js back end receives requests from the front end, interacts with third party systems and communicates with the MongoDB database. The use of separate frameworks for the front end and back end enables the development team of Digitomize to adapt these two systems independently, depending on the needs of growth.



B. Contest Aggregation

One of the main advantages of Digitomize consists in aggregating multiple sources of content or contest-related information in a single feed. It works using the respective APIs (Codeforces API and LeetCode GraphQL API), which help to show the titles of the ongoing or future contests as well as start/finish times, length, and difficulty level. Since there are differences between these platforms as regards the way they provide their information, the output will be unified and presented in one internal format.

As soon as there is a list of contests created with the help of aggregation, it becomes possible for users of the application to get a filtered list of these contests based on the criteria set by them.

C. Authentication and User Data

As regards the issue of user authentication in Digitomize, the platform uses tokens to authenticate its users, hashing passwords beforehand. Also, the JSON Web Tokens are issued to grant authorization after the user has entered his/her credentials. Thus, the user can log in with the help of email/password credentials or by signing up via Google. The features, such as portfolio editing or access to individual analytics, are available exclusively for logged-in users.

D. Portfolio Generation

As soon as a user adds their accounts to Digitomize, it begins creating a single, integrated portfolio based on LeetCode, Codeforces, and HackerRank. The portfolio displays a user's current rating, number of problems completed, history of contest participation, and other achievements. It is updated dynamically, so it always displays current information about users' status. Users may share a link to their portfolio publicly or download it as a PDF document, which proves to be quite useful when submitting job applications or internship requests.

E. Performance Analytics

In addition to merely presenting statistical data, Digitomize analyzes it using graphs and helps users understand the results. Users can see performance trends over time, observe their most productive times, and discover weaknesses in performance that need to be addressed. One thing that performance analysis does well is to detect some subtle patterns that would remain unnoticed without a graph. For example, it can show whether the user is stronger at completing tasks in shorter competitions and how their problem-solving skills have developed within different categories.

F. Notifications and Recommendations

One of the toughest challenges associated with competitive programming is being consistent. This issue can be tackled by notifications sent to users whenever a contest that might interest them is about to happen. Additionally, recommendations are provided based on the user's history of activity on the website. These prompts are expected to mitigate the time lag between intention and implementation, as they will motivate users to practice before getting out of practice, not after.

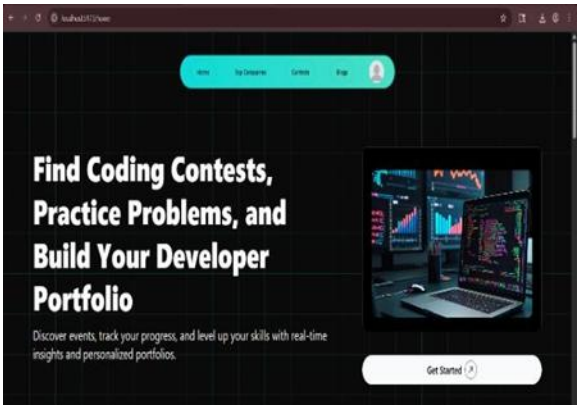
G. Technology Stack

React.js is used in the front-end development due to its component-oriented nature and its ability to render updates to the user interface when the underlying data changes. The back-end is developed using Node.js and Express.js to provide an efficient server architecture capable of handling multiple API calls. MongoDB is employed as the database to accommodate the flexible schema required to store heterogeneous data.

whether its fundamental characteristics operate as expected. These include the effectiveness of contests aggregation, portfolio creation, and the responsiveness of the analytics interface. In summary, all expectations for functionality have been

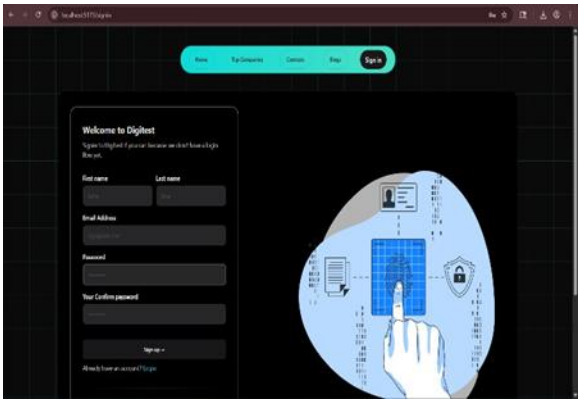
achieved by Digitomize through testing. The characteristics of each component are highlighted below.

1. Home Interface



The home page was created to inform about the functions of the platform clearly and concisely. The top navigation bar contains links to essential pages such as Home, Top Companies, Contests, and Blogs. Moreover, clear headings and brief descriptions make the navigation more user-friendly. Users receive suggestions on what actions can help achieve their goals through call-to-action buttons. The interface is fully responsive, which means that it operates perfectly on all screens regardless of size.

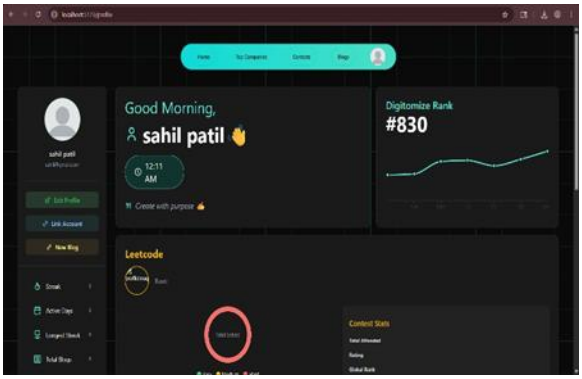
2. Authentication Module



IV. RESULTS

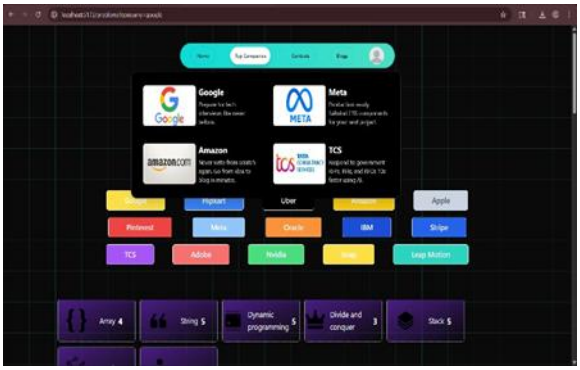
To validate the basic functionality of the platform, various scenarios were tested to see Both email and Google authentication work without any flaws. The password and token validation systems work perfectly, providing safe interaction with the platform. Unauthenticated users cannot gain access to such sensitive data as their portfolio or personalized recommendations. However, for registered users, access to these data is smooth without any obstacles.

3. Portfolio Module



The next step after linking their coding platform accounts is viewing a portfolio where the live data will be updated automatically without any need for manual effort from the user. The ratings, the number of problems solved by the user, contests attended, etc., are all updated automatically on a chart. This portfolio module is loved by many users since they can download it in a PDF file and/or share it directly. It's especially helpful for candidates looking for campus placements or applying for jobs.

4. Top Companies Module



This part categorizes the problems asked by coding companies like Google, Amazon, Meta, TCS, and others. Problems can be sorted first according to the organization and then filtered according to topics such as arrays, strings, dynamic programming, and so on. The purpose of doing this is so that it becomes easier for the user to solve only those kinds of problems that will possibly come up during their interviews.

V. CONCLUSION

In sum, Digitomize effectively solves the longstanding problem in the competitive

programming landscape by integrating several key elements into one tool: contest monitoring, performance metrics, and portfolio management. Instead of having to juggle different sources of information and compile everything manually, which is time-consuming and difficult, the platform performs all the necessary work by combining real-time information, generating portfolios for users, and providing timely reminders to keep users consistent. With a robust and up-to-date technology stack that includes React.js, Node.js, and MongoDB, Digitomize illustrates how well thoughtout API integration and good software design principles can create a coherent and productive experience for users. As for future development, Digitomize certainly has a number of features that might be added to increase its functionality. The addition of AI to the system and generation of individualized studying strategies based on a user's weaker areas is something that might prove to be quite useful, while integration with even more platforms, such as AtCoder and GeeksforGeeks, would increase the platform's appeal for those who use several different websites. Overall, with competitive programming being one of the mainstays of the technical world in education and recruitment, a solution like Digitomize seems quite promising indeed.

REFERENCES

[1] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education.

[2] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

[3] K. Chodorow and M. Dirolf, MongoDB: The Definitive Guide. O'Reilly Media.

[4] W. Pohl, "Competitive Programming Platforms and Their Impact on Learning," in Proc. 2019 ACM Conf. on Global Computing Education, 2019.

[5] I. S. Zinovieva, V. O. Artemchuk, A. V. Iatsyshyn et al., "The Use of Online Coding Platforms as Additional Distance Tools in Programming Education," Journal of Physics: Conference Series, ISSN: 1742-6596, 2021.

[6] M. Ibanez, A. Di-Serio, D. Villaran, and C. Delgado-Kloos, "The Impact of Gamification on Students' Learning, Engagement, and Behavior Based on Personality Traits in a Web-Based Programming Environment," Smart Learning Environments, ISSN: 2196-7091, 2019.

[7] T. T. Yuen, "Competitive Programming in Computational Thinking," Wiley, 2023.

[8] M. A. Habib et al., "REST-Based Aggregated Data Integration," Sensors, vol. 22, no. 15, ISSN: 1424-8220, 2022.

[9] K. Vinothini et al., "MERN Stack Web Portals," in Proc. International Conference on Intelligent Computing and Technology (ICICT), 2022.

[10] M. Billah et al., "LLMs and Programming Platforms," in Proc. ACM Int. Symposium on Empirical Software Engineering and Measurement (ESEM), 2024.

[11] A. Seth et al., "MARCREST vs GraphQL: A Comparative Analysis," Springer, 2024.

[12] R. Gm et al., "Digital Recommendation Systems for Personalized Learning," IEEE Access, vol. 12, ISSN: 2169-3536, 2024.