



CODE PROFICIENCY PLATFORM

G Mahesh Challari¹, K.Naga Lakshmi², P.Aashritha³, M.Chanti Kumar⁴, K.Shiva Kumar⁵, D.Srikanth⁶
Assistant Professor, Department of Computer Science and Engineering¹
Student, Department of Computer Science and Engineering^{2,3,4,5,6}
Sree Dattha Institute of Engineering and Science, Sheriguda, Telangana. ^{1,2,3,4,5,6}

ABSTRACT :

In this ongoing Research to Practice paper, we investigate the integration of a Coding Assessment Portal with an automatic grading system, implemented across two consecutive quarters for a large-scale introduction to programming course. Initially, our approach employed an on-demand standalone automatic grading system alongside a separate assignment submission portal on Canvas. Following thorough evaluation and specific student feedback, we integrated the assignment submission portal with the autograder system (Stepik) to facilitate real-time, objective assessment of assignments. The key outcome of this integration was a significant 20.5% increase in the class average of assignment scores, despite most test cases being hidden during evaluation. Notably, our approach also led to a notable reduction in the DFW (Drop/Fail/Withdraw) rate, decreasing it from 46% to 12.5%, and a substantial 22% increase in the passing rate among female students. Moreover, in the second iteration of the course, elective students demonstrated comparable or better performance than those taking it as a requirement. Additionally, the autograder system proved beneficial in enhancing students' code quality. Furthermore, our research addresses concerns regarding source code plagiarism in computer science education. We developed a plagiarism detection tool tailored for Python programming languages, utilizing a three-step process: tokenization, N-Gram representation, and comparison via the Greedy String Tiling method. Our tool achieves a response time of one minute for fifty source code documents, each consisting of seventy-five lines of code (LOC). Feedback from educators who utilized our tool in graduate computing programs has been overwhelmingly positive, reporting accuracy rates of 99%. This tool effectively supports educators in monitoring and maintaining academic integrity by identifying instances of plagiarism in student code submissions. We firmly believe that our approach not only evaluates students' true abilities but also significantly aids educators in ensuring fair assessment practices and fostering a conducive learning environment.

Index Terms: Coding Assessment Portal, automatic grading system, real-time assessment, DFW rate, source code plagiarism, tokenization, N-Gram representation, Greedy String Tiling, academic integrity, Python programming.

DOI Number: 10.48047/nq.2024.22.4.nq24009

NeuroQuantology 2024; 22(4):80-85

I. INTRODUCTION

Welcome to our cutting-edge web application designed for programming enthusiasts! Our platform provides a seamless experience where users can sign up, log in, and engage in coding challenges using Java or Python programming languages. Once users submit their code, our

robust backend infrastructure compiles and executes it in a secure environment.

We prioritize user feedback and performance evaluation, offering real-time grading based on the correctness and efficiency of the submitted programs. Our platform supports a variety of



features to enhance the learning and coding experience, including automated testing, syntax highlighting, and detailed performance analysis. Whether you're a seasoned developer honing your skills or a novice exploring the world of programming, our application offers a supportive environment to test, refine, and showcase your coding prowess. Join us on this journey of learning and mastery in the realm of software development.

PURPOSE:

The purpose of on-line test simulator is to take online test in an efficient manner and no time wasting for checking the paper. The main objective of on-line test simulator is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. For students they give papers according to their convenience and time and there is no need of using extra thing like paper, pen etc.

II.LITERATURE SURVEY

Related work in automated grading systems and plagiarism detection tools has significantly advanced the landscape of programming education. Automated grading systems, akin to the described platform, streamline the assessment process by integrating with learning management systems (LMS) or functioning as standalone tools. These systems leverage automated testing, syntax checking, and performance analysis to provide objective feedback on code submissions in real-time. This integration not only enhances grading efficiency and consistency but also supports timely feedback to students, thereby improving learning outcomes. Concurrently, plagiarism detection tools employing techniques such as tokenization, N-Gram representation, and similarity measures like Greedy String Tiling play a crucial role in maintaining academic integrity. These tools aid educators in identifying instances of code similarity across submissions, ensuring fair evaluation and reinforcing ethical standards in programming education. Research indicates that such technologies not only boost assignment scores and reduce withdrawal rates

but also promote equitable learning opportunities by narrowing performance gaps among student cohorts. As computing technologies evolve, ongoing advancements in cloud computing and secure execution environments further enhance the scalability and reliability of these systems, supporting diverse programming languages and enriching the educational experience for both educators and learners.

III.EXISTING SYSTEM :

In introductory Computer Science (CS) courses, accurate, meaningful, timely, and efficient feedback on programming assignments is essential for the learning experience. The timely assessment of assignments is also crucial from the teacher's perspective as it provides feedback on how well learning goals are being achieved [1]. Users can write and edit programs in Java or Python directly within the application's programming environment, which is equipped with essential features such as syntax highlighting, code completion, and formatting tools to facilitate efficient coding practices. Upon submission, the system integrates with compilers and interpreters to execute user-written code in a secure sandbox environment. Automated testing mechanisms evaluate the correctness and efficiency of the programs, providing real-time feedback and grades based on predefined criteria such as output accuracy and adherence to coding standards. The system also monitors performance metrics like runtime duration and memory usage to assist users in optimizing their code.

DRAWBACKS:

- One disadvantage is potential limitations in handling complex programming scenarios or unusual edge cases.
- Another challenge could be ensuring consistent and fair automated grading across diverse coding styles and approaches.

IV.PROPOSED SYSTEM :

In this project we are designing web application



where users can sign up with the application and then login and can write programs in JAVA or PYTHON and then application will compile and execute code and based on programs correctness application will give grade to the user.

ADVANTAGES:

Advantages include providing hands-on programming practice with immediate feedback and enabling scalable assessment of coding proficiency.

Enables scalable assessment of coding skills, beneficial for both learners and evaluators.

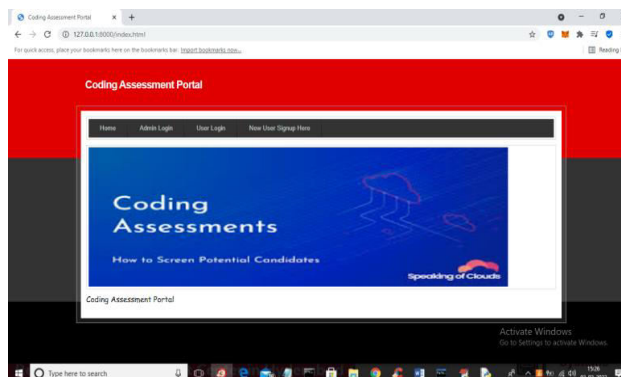
V.IMPLEMENTATION

MODULES:

To implement this project we have designed following modules

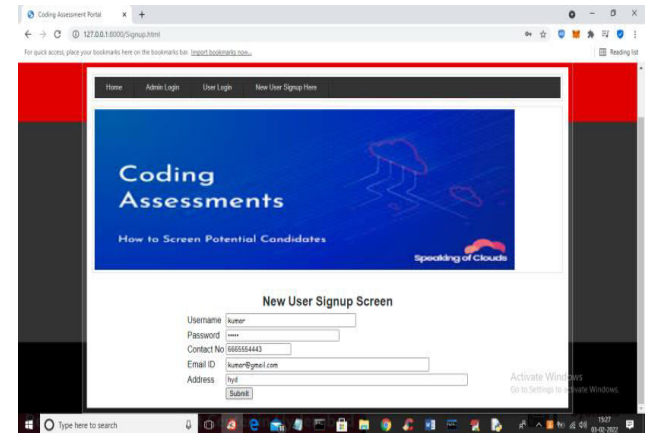
- 1) New User Signup: using this module user can signup with the application
- 2) User Login: using this module user can login to application
- 3) Python Assessment: using this module user can write python program and then application will compile and run code and based on correctness application will give grade
- 4) Java Assessment: using this module user can write java program and then application will compile and run code and based on correctness application will give grade
- 5) Admin Login; admin can login to application using username and password as 'admin' and 'admin' and after login admin can view all registered user data

VI.RESULTS:



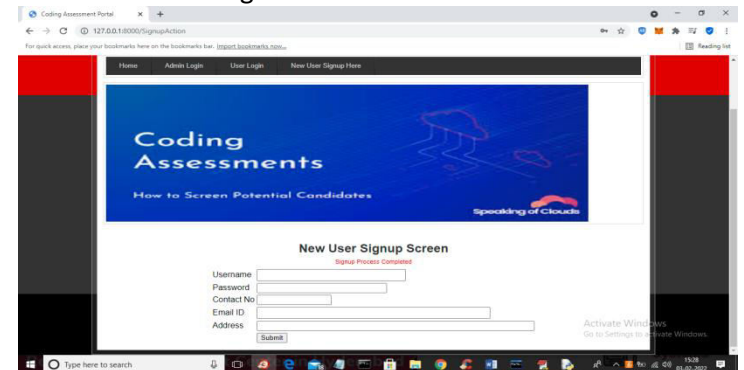
In above screen click on 'New User Signup Here' link to add new user and to get below screen

In above screen user

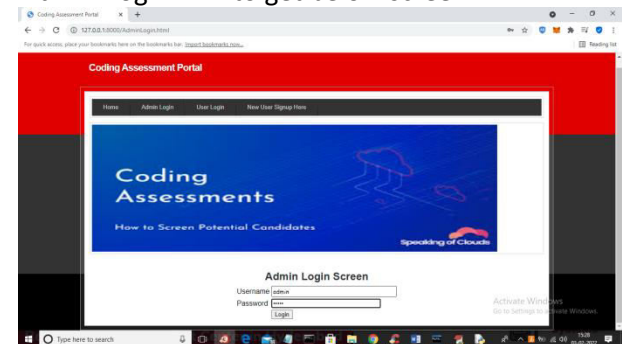


is entering signup details and then press 'Submit' button to get below screen

82

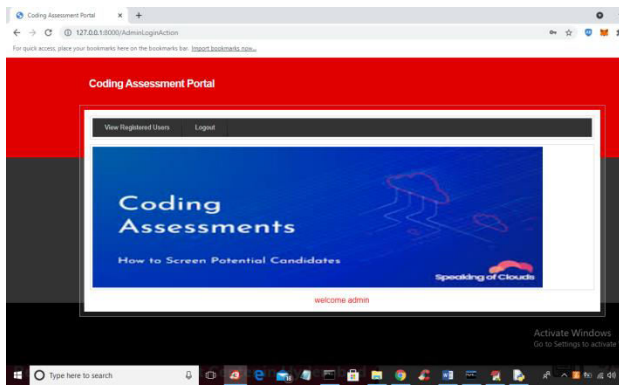


In above screen in red colour text we can see signup process completed and now click on 'Admin Login' link to get below screen

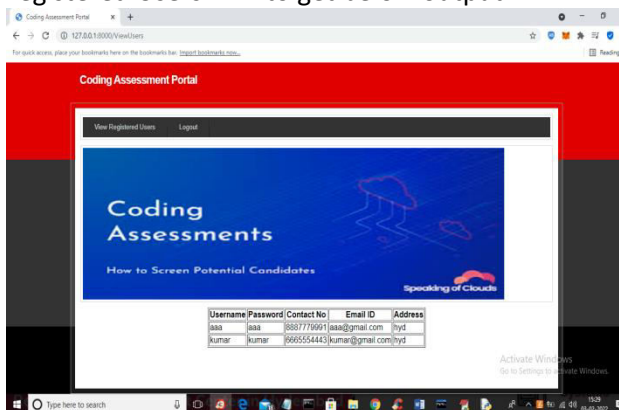


In above screen admin is login and after login will get below screen

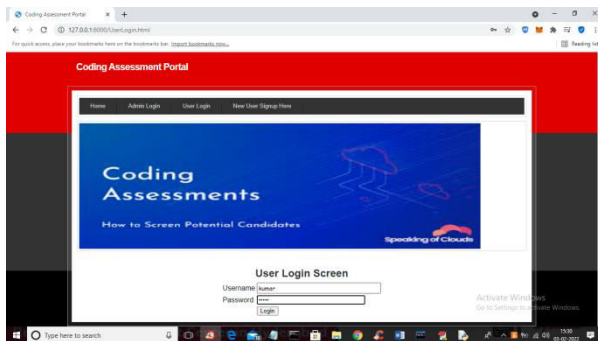




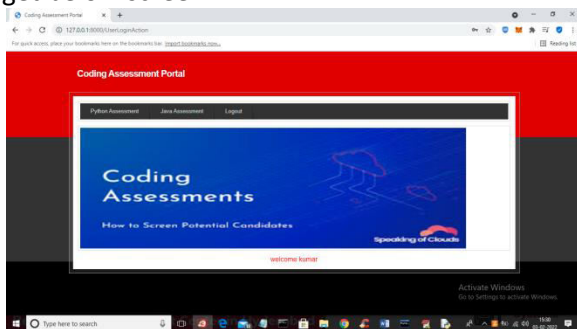
In above screen admin can click on 'View Registered Users' link to get below output.



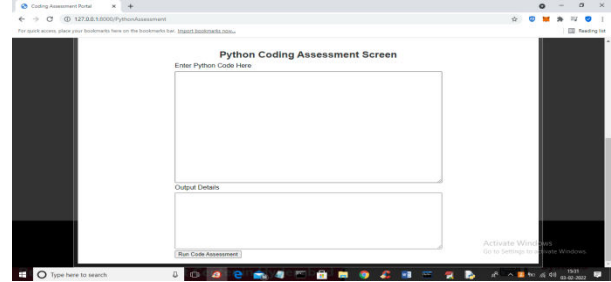
In above screen admin can view all registered user details and now logout and login as kumar user



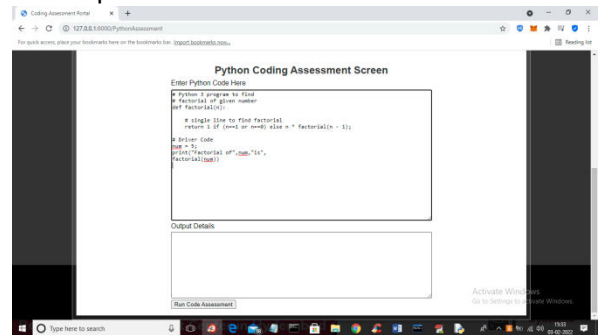
In above screen user is login and after login will get below screen



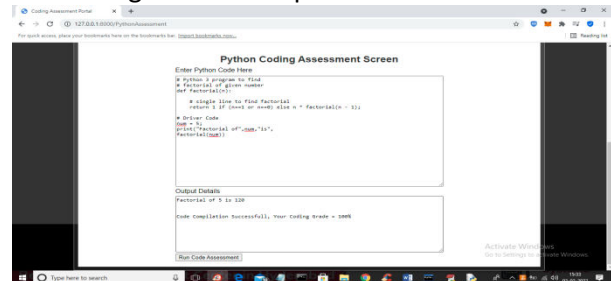
In above screen user can click on 'Python Assessment' link to get below screen



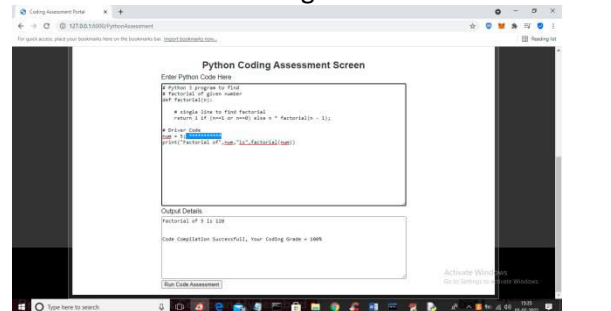
In above screen in first text area you can enter python code and in second text area you can see output



In above screen I entered some factorial program and now press 'Run Code Assessment' button to get below output



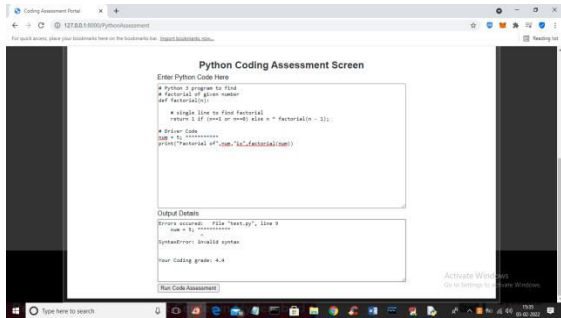
In above screen in second text area we got output as 'Factorial of 5 is 120' and we got message as Code Compilation Successful and got grade as 100% and now put some error in code and then execute again



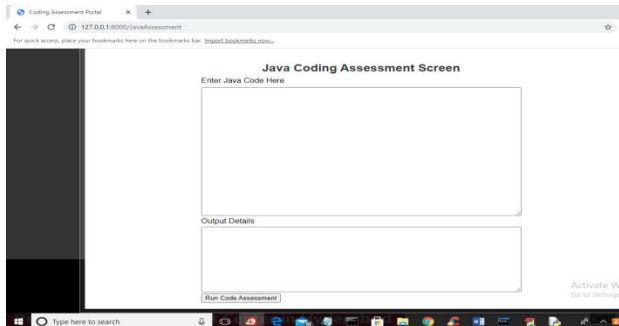
In above screen in blue colour text you can see I



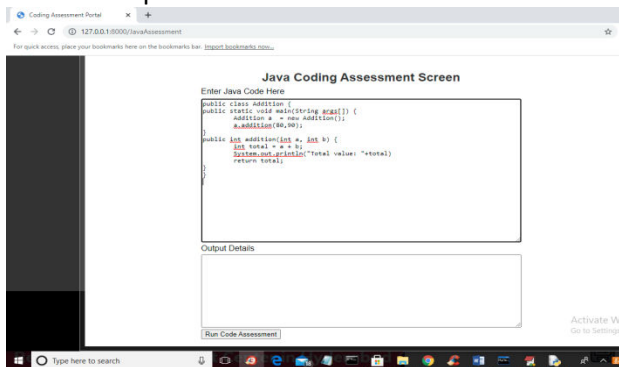
added some garbage * symbols which are not valid and press button to get below compilation error



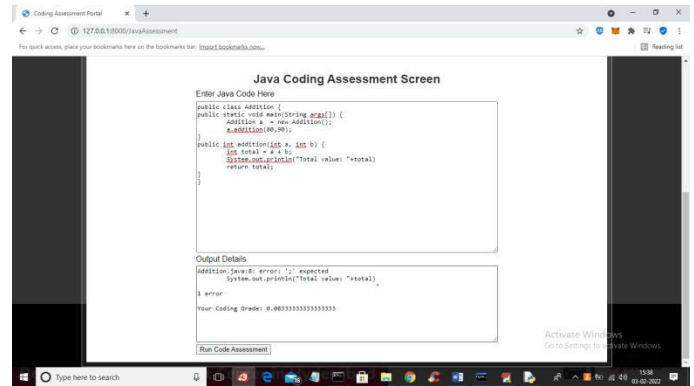
In above screen in second text area we got error details and we got grade as 4.4% and similarly you can test other programs also. Now click on 'Java Assessment' link to get below screen



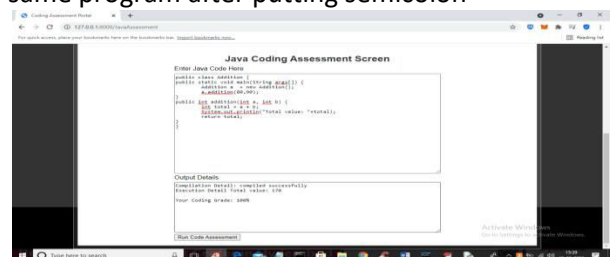
In above screen in first text area you can enter java program and in second text area u will see error or output details



In above screen I entered some addition of 2 numbers program and press button to get below output



In above screen I got error as I forgot to put ';' after System.out. statement and now I will run same program after putting semicolon



In above screen program got compiled and got addition output as 170 and got grade as 100% as program compiled successfully

VIII.CONCLUSION

The On line Coding Assessment Prortal is developed using Python and Mysql fully meets the objectives of the system for which it has been developed. The system has reached a steady state whereall bugs have been eliminated. The system is operated at a high level of efficiency and all the teachers and user associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification.

proposed and enforced a plagiarism checker powered by AI, there are some notable limitations, prominent among which was the small corpus size. Larger data sets would be needed to test the scalability of the model and there are intentions to explore this in unborn workshop. Likewise, cross lingual plagiarism check is also an avenue to explore in the future, as by investing NLP, our model is high formulti-lingual support. If the model is exposed to data in multiple languages, it should be suitable to understand the relationship between words across colourful languages.



IX.FUTURE ENHANCEMENT

The scope of this project is extensive, particularly in comparison to traditional, manually administered exams. Firstly, it can be utilized not only in educational institutions but also in the corporate world, providing a versatile tool for both academic and professional assessments. Additionally, as a web-based application, it offers the convenience of being accessible from any location at any time, thereby eliminating geographical constraints for users. This flexibility is further enhanced by the fact that there is no requirement for the examiner to be physically present when the candidate takes the test, allowing for asynchronous testing and evaluation. These features collectively make the system a highly adaptable and efficient solution for modern assessment needs.

X.REFERENCES

- [1] Ihantola, Petri, Tuukka Ahoniemi, Ville Karavirta, and Otto Seppala. "Review of recent systems for automatic assessment of programming assignments." In Proceedings of the 10th Koli calling international conference on computing education research, pp. 86-93. ACM, 2010
- [2] Nordquist, Pete. "Providing accurate and timely feedback by automatically grading student programming labs." *Journal of Computing Sciences in Colleges* 23, no. 2 (2007): 16-23.
- [3] DeNero, John, and Stephen Martinis. "Teaching composition quality at scale: human judgment in the age of autograders." In Proceedings of the 45th ACM technical symposium on Computer science education, pp. 421-426. ACM, 2014.
- [4] "CRA Releases Report on Surge in Computer Science Enrollments" CRA. CRA, 24 February 2017. Web. 1 May 2018.
- [5] Danutama, Karol, and Inggriani Liem. "Scalable autograder and LMS integration." *Procedia Technology* 11 (2013): 388-395.3. [6] Helmick, Michael T. "Interface-based programming assignments and automatic grading of java programs." In ACM SIGCSE

- Bulletin, vol. 39, no. 3, pp. 63-67. ACM, 2007.0.
- [7] FERNANDO, Jordan, and MM LIEM. "Components and Architectural Design of an Autograder System Family." *Olympiads in Informatics* 8 (2014).
 - [8] Kunchala, Ashrita, and Maneesh Gunnala Ranga Rao. "Java Auto Grader." (2016).
 - [9] Davuluri, Prathibha, and Pranavi Reddy Madadi. "Moodle Java Autograder." (2016).
 - [10] Zimmerman, Daniel M., Joseph R. Kiniry, and Fintan Fairmichael. "Toward instant gradeification." In *Software Engineering Education and Training (CSEE&T)*, 2011 24th IEEE-CS Conference on, pp. 406-410. IEEE, 2011.

